

Sistema experto de armonización en contrapunto severo a dos voces.

(1ª y 2ª especie)

Apellidos: Padilla Martín-Caro
Nombre: Víctor
E-Mail: VPADILLA@inicia.es
Tfno: 91/8744305. (639) 929827
Asignaturas Doctorado (98/99)

- Inteligencia artificial (fundamentos)
- Inteligencia artificial (aplicaciones)

1.- Introducción sobre la IA y la música.

El término “inteligencia artificial y música” fue acuñado por Curtis Roads, referido a las investigaciones que se centran en la música como un proceso cognitivo, o conjunto de actividades, modeladas con la ayuda de programas informáticos. Los comienzos en esta disciplina datan de finales de los 60 que intentaban responder a cuestiones que tenían que ver con la naturaleza del conocimiento musical, tales como:

- ¿Qué es el conocimiento o la inteligencia musical?
- ¿Es el conocimiento musical una capacidad humana autónoma o una colección de capacidades relacionadas históricamente?
- La capacidad musical, ¿es diferente de otras capacidades, como el lenguaje?
- La capacidad musical es invariable por encima de periodos históricos y culturas.
- ¿Cómo se relacionan el conocimiento musical y la música práctica?

Marvin Minsky [Minski 86] defiende que se puede modelar el comportamiento humano en las actividades musicales y por tanto crear sistemas inteligentes de comportamiento musical. Por lo tanto parece razonable, al menos como punto de partida, adoptar métodos desarrollados en IA para el modelado de actividades musicales.

Sin embargo hay también diferencias importantes entre la investigación tradicional en IA y la investigación en IA y música. Una diferencia tiene que ver con la propia naturaleza de las actividades musicales, porque, después de todo, ¿qué es la música?. Muchas tendencias en IA, por ejemplo en el procesado del lenguaje natural, se centran en modelar situaciones del mundo real. Hacer música, para Minski y Laske por ejemplo, es análogo a contar una historia, pero ¿cual es el referente de esa historia?. No está claro si el dominio de referencia musical es emocional, perceptivo, intelectual, psicológico o una mezcla de éstos.

La mayoría de las investigaciones en IA enfatizan la importancia de la representación del nivel de conocimiento y su conceptualización. Ambas son, por supuesto, abstracciones del lenguaje natural. La idea principal de esta aproximación es que normalmente no hay una única base de conocimiento que describa los fenómenos modelados; más bien un fenómeno se puede describir en varios niveles de conocimiento, cada uno reflejando una perspectiva diferente. Estos niveles de conocimiento pueden estar o no relacionados. La clasificación del conocimiento separado en niveles concuerda bien con los múltiples puntos de vista empleados en describir el conocimiento musical y con la variedad de actividades musicales conocidas.

Distintas vías de investigación en el campo de la IA y la música son: composición musical, análisis musical, interpretación, percepción, aprendizaje, etc. Para este trabajo nos centraremos en el campo del análisis musical, especialmente en los trabajos realizados por K. Ebcioglu y J. Maxwell. sobre música barroca.

Análisis musical.

El análisis musical es una tarea mental difícil, y su modelado es uno de los retos más interesantes en el campo de la IA y la música. Los primeros ejemplos en este campo son los proyectos para construir un índice temático de los trabajos de Josquin y determinar el autor de una serie de piezas musicales [Lincoln 70]. Otros ejemplos son programas que realizan análisis armónicos de los corales de Bach [Winograd 68], análisis Schenkerianos [Smoliar 80, Kassler 75, Snell 79].

El método quizás más ambicioso de aproximación a un estilo musical ha sido escribir programas que intenten generar piezas en este mismo estilo. Escribiendo un algoritmo para generar nuevas piezas similares a unas dadas, los investigadores esperan obtener un entendimiento más detallado de la música en ese estilo y eliminar los vagos e incompletos estados verbales que son difíciles de evitar en una aproximación tradicional al análisis musical. Un ejemplo de este tipo es el análisis por síntesis lo proporcionan el sistema experto creado por Ebcioglu para la armonización de corales de Bach, aunque esta tecnología de análisis por síntesis está todavía lejos de producir piezas nuevas y complejas como las que el compositor original pudiera haber escrito.

2.- Introducción a la implementación

Este trabajo pretende ser, a pequeña escala, un sistema experto que sea capaz de realizar una segunda melodía a una dada (cantus firmus) basándose en las reglas del contrapunto severo a dos voces. Se trata de un sistema basado en reglas donde a la entrada del mismo estaría la melodía a armonizar (cantus firmus) y a la salida obtenemos las distintas posibilidades de armonizar esta melodía a dos voces, jerarquizadas en función de las reglas introducidas (reglas armónicas del contrapunto).

La implementación final del sistema ha sido en Visual Basic. Se ha escrito el código de la primera y la segunda especie del contrapunto severo. Posibles ampliaciones de este trabajo pasarían por implementar la tercera y la cuarta especie, así como mejorar la eficiencia del algoritmo por medio de funciones en lenguaje C que acelerasen el proceso de cálculo, compiladas en ficheros *.dll, que fuesen invocados desde el programa principal.

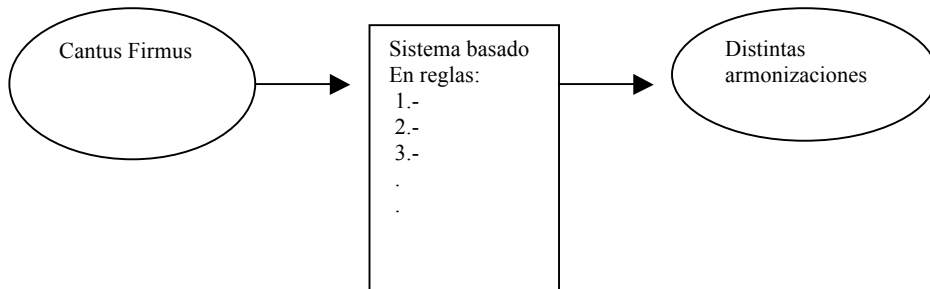


Fig. 1

3.- Contrapunto severo. (Especies)

El contrapunto severo es un procedimiento escolástico para iniciarse en el aprendizaje del contrapunto, práctica usual en la rama de composición de los conservatorios de música occidentales. Esta forma de iniciarse en la escritura contrapuntística tiene su origen en la obra del siglo XVII “Gradus ad Parnassum” de J.J. Fux. Fux, y sus seguidores, hacen una distinción del contrapunto en cinco especies sobre una melodía o cantus firmus fija en redondas. Estas cinco especies son las siguientes [Schoenberg 63]:

1.- Primera especie: Adición de una segunda voz al cantus firmus también en redondas (nota contra nota). Esto nos recuerda las primeras composiciones contrapuntísticas medievales de los siglos IX y X, el organum paralelo y el discantus, a pesar de que las reglas armónicas de conducción de voces están basadas en los procedimientos renacentistas del siglo XVI. Nosotros seguiremos las reglas de la primera especie (en concreto de los tratados de Schoenberg, Forner y Calés), pero para conseguir una armonización acorde a los tratados medievales (Musica Enchiriadis), sólo habría que modificar algunas reglas.

Voz 1	W	W	W	W	W	W
C.F.	W	W	W	W	W	W

2.- Segunda especie: Adición al cantus firmus de una segunda voz en blancas. La segunda voz está escrita al doble de velocidad de la primera. Por cada nota del cantus firmus tenemos dos de la nueva melodía.

Voz 1	h	h	h	h	h	h	h
C.F.	w	w	w	w			

3.- Tercera especie: Adición al cantus firmus de una segunda voz en negras (cuatro veces más rápida). Recuerda a los organum melismáticos de Leonin y Perotín de los siglos XII y XIII, con la salvedad de la armonía renacentista empleada.

Voz 1	q	q	q	q	q	q	q	q	q	q	q	q	q	q	q	q
C.F.	w		w		w		w		w							

4.- Cuarta especie: Adición al cantus firmus de una segunda voz en blancas sincopadas. Las dos voces van a la misma velocidad (redondas) pero desplazadas en tiempo un pulso de blanca.

Voz 1		h	----	h		h	----	h		h	----	h		h
C.F.		w		w		w		w						

5.- Quinta especie: Consiste en combinar los procedimientos anteriores. En la voz superior puede haber redondas, negras, blancas y sincopas de blanca.

Voz 1	h	q	q	q	q	h	----	h	h
-------	---	---	---	---	---	---	------	---	---

C.F. w w w w

4.- Modelado del conocimiento.

En este apartado se explica la base de conocimiento del sistema tonal temperado y de la primera y la segunda especie del contrapunto simple a dos voces. Esta base de conocimiento ha sido extraída de tratados tradicionales de armonía y contrapunto. [Schoenberg 63], [Schoenberg 22], .

4.1.- Sistema tonal temperado:

- El sistema tonal occidental, empleado tradicionalmente en Europa desde el barroco hasta nuestros días, está formado por 12 divisiones iguales de la octava (logarítmicamente hablando). De do a DO tenemos, empleando la escala de sostenidos, las siguientes 12 notas:

do, do#, re, re#, mi, fa, fa#, sol, so#, la, la#, si, DO

- Las escalas empleadas son la escala mayor y la escala menor. Para nuestro trabajo nos vamos a quedar sólo con la escala mayor de Do (Do mayor) que está formada por las siguientes notas:

...do, re, mi, fa, sol, la, si, DO, ...

-Una escala mayor está formada por tonos y semitonos agrupados de la siguiente manera:

tono, tono, semitono, tono, tono, tono, semitono.

Por lo tanto:

do – re = tono

re – mi = tono

mi – fa = semitono

fa – sol = tono

sol – la = tono

la – si = tono

si- do = semitono

- Los intervalos armónicos que podemos tener a dos voces (notas sonando simultáneamente) son:

Unísono:	la misma nota en las dos voces.	Ej: do-do
Segunda menor:	1 semitono	Ej: mi-fa
Segunda mayor:	2 semitonos	Ej: do-re
Tercera menor:	3 semitonos	Ej: re-fa
Tercera mayor:	4 semitonos	Ej: do-mi
Cuarta disminuida	4 semitonos	Ej: do-fa(b) (no empleado en do
	mayor, en severo)	
Cuarta justa	5 semitonos	Ej: do-fa
Cuarta aumentada	6 semitonos	Ej: fa-si
Quinta disminuida	6 semitonos	Ej: si-fa
Quinta justa	7 semitonos	Ej: do-sol
Quinta aumentada	8 semitonos	Ej: do-sol(#) (no empleado en do
	mayor, en severo)	
Sexta menor	8 semitonos	Ej: la-Fa
Sexta mayor	9 semitonos	Ej: do-la
Séptima menor	10 semitonos	Ej: re-Do
Séptima mayor	11 semitonos	Ej: do-si
Octava justa	12 semitonos	Ej: do-Do

4.1.- Primera especie:

Reglas y consejos:

1.- A cada nota del cantus firmus (CF) se le ha de añadir una nota consonante. La adición de notas consonantes sólo está condicionada por las exigencias de la correcta conducción de las voces.

2.- Son consonancias (o intervalos consonantes) las siguientes relaciones entre dos notas:

- a) 1, la primera, que no es un intervalo, porque ambas voces cantan o tocan el mismo sonido.
- b) 8, la octava, tanto por encima como por debajo de una nota.
- c) 5, la quinta –que ha de ser justa- tanto por encima como por debajo de una nota, y no así la quinta disminuida que es una disonancia.
- d) 3, la tercera, mayor o menor, tanto por encima como por debajo de la nota.
- e) 6, la sexta, mayor o menor, tanto por encima como por debajo de la nota.

3.- En la primera especie no se debe utilizar ningún intervalo disonante. Son disonancias las siguientes relaciones entre dos notas:

- a) 2, la segunda, tanto por encima como por debajo de una nota
- b) 7, la séptima, tanto por encima como por debajo de una nota.
- c) 9, la novena, tanto por encima como por debajo de una nota (de hecho, una novena es una segunda más una octava)
- d) 4, la cuarta, tanto por encima como por debajo de una nota (la cuarta es una consonancia imperfecta, pero a dos voces debe ser considerada y tratada como disonancia).
- e) La cuarta aumentada y su inversión, la quinta disminuida, intervalos que en la escala mayor aparecen entre las notas cuarta y séptima y viceversa.
- f) Cualesquiera otros intervalos aumentados y disminuidos que pudieran aparecer posteriormente, tales como los de la escala menor.

4.- Todo intervalo conserva su calificación de consonancia o disonancia, respectivamente, aún cuando dicho intervalo esté ampliado en una o más octavas.

Voces y claves:

5.- Las voces que se han de emplear son las mismas que se utilizan en el actual coro mixto: es decir, soprano, contralto, tenor y bajo.

6.- Para los ejercicios se han de emplear siempre dos voces vecinas. Por ejemplo, si el CF está en el contralto, la voz sobre él quedará en la soprano.

Conducción melódica de las voces:

7.- Las voces han de ser melodiosas o, al menos, no amelódicas. Para ello deben seguirse los siguientes consejos:

- a) Debe evitarse todo intervalo aumentado o disminuido. Además como sólo se deben emplear las notas naturales de la escala diatónica, quedarán asimismo excluidas las progresiones cromáticas.
- b) El intervalo de cuarta aumentada entre las notas cuarta y séptima debe ser evitado. Lo mismo con su inversión, la quinta disminuida.
- c) Se han de evitar aquellas progresiones en las que una o más notas quedan insertas entre las notas cuarta y séptima o entre séptima y cuarta de la escala (tritonos compuestos).

- d) Hay que evitar estrictamente saltos con los intervalos disonantes de séptima, novena, undécima, etc.
- e) Se deben evitar también estos intervalos cuando estén producidos por dos saltos sucesivos, en la misma dirección, de intervalos permitidos. Se transforman entonces en saltos disonantes compuestos.
- f) El movimiento en una sola dirección no debe continuar demasiado tiempo. Después de un salto debe evitarse proceder en la misma dirección, incluso por grados conjuntos. En lugar de eso, el salto ha de equilibrarse mediante un movimiento en dirección opuesta, sea por grados conjuntos o por salto.
- g) Debe evitarse el salto de sexta, porque en muchos casos produce octavas o quintas ocultas. Los teóricos del contrapunto tradicional permitan la sexta menor ascendente pero es mejor evitarla también. Por supuesto que en una emergencia, si se evitan paralelas incorrectas, puede emplearse tanto la sexta mayor como la menor, ya ascendentes ya descendentes.
- h) Los acordes arpegiados han de evitarse tanto como sea posible, por dos razones. En primer lugar, porque pueden expresar una armonía y restringir por tanto, el libre movimiento de las otras voces. En segundo término, porque dan frecuentemente la impresión de un acompañamiento en estilo homofónico.
- i) A menudo se produce monotonía por la repetición frecuente de una nota o de una sucesión de varias notas, y por permanecer demasiado tiempo en el ámbito de una cuarta o una quinta.
- j) Las repeticiones secuenciales, es decir, las repeticiones de una sucesión de notas sobre otro grado (una segunda, una tercera, una cuarta, etc., más arriba o más abajo) también deben ser evitadas.
- k) El salto de octava puede utilizarse en caso de emergencia si, por ejemplo se hubiese rebasado el ámbito de la voz o si, por otra parte, las voces se hubiesen acercado o alejado demasiado. No se permite salto alguno mayor que la octava.
- l) Una buena melodía ha de mezclar los saltos con el movimiento por grados conjuntos y no proceder demasiado tiempo en una misma dirección.

Independencia de las voces

8.- Las voces deben encontrarse en determinados puntos en armonías comprensibles y que, juntas, deben expresar claramente la tonalidad. Por lo demás han de ser tan independientes como resulte posible. Al escribir para dos o más voces pueden darse tres tipos de movimiento: paralelo, contrario y oblicuo.

- a) El movimiento paralelo, es decir, el movimiento en la misma dirección, es el tipo de conducción de las voces menos independientes. Si se prolonga durante un tiempo considerable, una de las voces ofrecerá el aspecto de

“sombra”. Pero el movimiento paralelo ocasional no puede evitarse, como tampoco destruye necesariamente la independencia.

- Las octavas y quintas paralelas explícitas (el movimiento paralelo de una octava a otra octava o de una quinta a otra quinta) deben ser estrictamente evitadas.
- Las llamadas “octavas de trompa” y “quintas de trompa” son permitidas, sin embargo, en el contrapunto simple a dos voces han de evitarse todas las demás quintas y octavas paralelas ocultas.
- Las terceras y sextas paralelas no son nunca incorrectas, aunque, como se dijo anteriormente, el movimiento paralelo no debe utilizarse demasiado tiempo en cada ocasión.

b) El movimiento contrario es aquel en que las voces se mueven en direcciones opuestas. Produce mayor grado de independencia que el movimiento paralelo.

c) El movimiento oblicuo es aquel en que una voz se mueve mientras la otra permanece estacionaria. Este tipo de movimiento no puede emplearse en la primera especie a menos que una de las voces repita nota. Ofrece al menos independencia rítmica. En toda pieza de cualquier longitud hallaremos presentes los tres tipos de movimiento: paralelo, contrario y oblicuo.

9.- La independencia se reduce si se encuentran con demasiada frecuencia en primeras u octavas, incluso por movimiento contrario. La primera y la octava deben reservarse para el comienzo, con el fin de expresar inequívocamente la tonalidad, y para el final, para confirmarla.

10.- Debe evitarse el cruzamiento de las voces. Aunque el cruce no es incorrecto, resulta por lo general un camino demasiado fácil para tratar los problemas.

11.- La nota penúltima se elegirá siempre de las pertenecientes al acorde de dominante (V) o de las del VII grado. Si es posible, debe aparecer aquí la nota sensible. La voz inferior puede usar también la fundamental de la dominante, pero nunca la fundamental de IV que coincide con la quinta disminuida de VII.

4.2.- Segunda especie:

Consejos y directrices

12.- Si se divide una redonda en dos (o en cuatro u ocho), la primera de cada par de notas se convierte en un tiempo acentuado o fuerte, mientras que la segunda queda sin acento, como tiempo débil. Si la redonda se divide en tres (o en seis), la primera de cada tres notas constituye un tiempo fuerte acentuado, en tanto que las dos que siguen son tiempos débiles sin acentuar. El primer tiempo de todo compás es el que lleva el acento principal y los tiempos fuertes del interior del compás quedan un poco menos acentuados.

13.- Hay una diferencia en el tratamiento de los tiempos fuertes y débiles en relación con el uso de disonancias y consonancias. Las consonancias (primera,

octava, quinta, tercera, sexta) pueden usarse libremente en tiempos fuerte y débiles. Las disonancias en tiempos débiles y con las siguientes reglas:

- a) Como nota de paso: es una disonancia en una fórmula de tres notas. La primera y la última de estas tres notas deben ser consonancias a distancias de tercera y han de estar situadas en tiempos fuertes. La nota que queda en medio de estas dos consonancias situada en tiempo débil, es una disonancia. La fórmula completa forma parte de una línea en escala que sube o baja.
- b) Como floreo: es una disonancia en una fórmula de tres notas. La primera y la última son iguales. La nota de en medio está a distancia de segunda.

La conclusión en la segunda especie

14.- Si el CF acaba con 2-1, la penúltima nota de la voz superior sólo puede ser la sensible (séptima nota de la escala mayor). Para el penúltimo compás sólo queda una posibilidad: de la sexta nota a la séptima y a la octava.

15.- La primera nota de la voz superior no debe ser otra que primera, octava, quinta o tercera. La sexta no expresaría la tonalidad de forma clara.

5.- Reflexiones compositivas.

El compositor o estudiante de contrapunto, no emplea reglas sin más y las aplica, sino que busca caminos interesantes, estima posibilidades de realización, se encuentra con situaciones no satisfactorias y retrocede (borra lo realizado). También hay que tener en cuenta que no todas las reglas tienen el mismo peso y que hay que hacer jerarquías. Por ejemplo:

- Regla x: Las quintas justas paralelas entre las voces están prohibidas.

Si establecemos un camino y nos encontramos dos quintas paralelas entre las voces, este camino será incorrecto y no será necesario seguir evaluando. Por tanto, podemos establecer un procedimiento de poda. [Mira 95]. Será inútil seguir expandiendo esta rama porque va a ser una armonización incorrecta.

-Regla y: Hay que evitar movimientos melódicos monótonos. Este camino debe ser penalizado en un determinado grado, pero no eliminado, porque nos puede conducir a continuación a una solución satisfactoria de la armonía. Para este tipo de reglas, parece lógico emplear la lógica difusa. [Mira 95].

Además de evaluar nota por nota, el compositor levanta la vista del papel y evalúa lo ya escrito y decide en función de lo ya hecho. Puede que algo no le guste y decida volver atrás. También puede darse el caso de que lo escrito sirva de “inspiración” a decisiones posteriores. Por tanto, en todo momento y en cualquier tipo de composición, parece necesario establecer un “back-traking” que nos permita seguir adelante [Ebcioglu, 92]. Esta evaluación debe realizarse a varios niveles:

- Evaluar la nota siguiente para evitar faltas.
- Evaluar, dentro de las posibles notas siguientes, aquellas que conduzcan a una mejor realización.
- Comparar lo hecho anteriormente con las nuevas posibilidades para realizar estimaciones [Marsella 92].
- Evaluar todo lo realizado. Si no llega a un determinado umbral, seleccionar otro camino.

6.- Sistema y algoritmos empleados.

6.1.- Primera especie

El sistema implementado consta de una entrada formada por un cantus firmus de 6 notas y como salida las distintas posibilidades de armonizarlo en primera y segunda especie. El hecho de que el CF tenga sólo 6 notas nos permite que el tiempo de cálculo no sea excesivo a la vez que este CF y sus posibles soluciones puedan ser vistas “de una vez”.

El método de trabajo es muy simple en el caso de la primera especie:

- 1.- Tomamos el CF
- 2.- Establecemos todos los caminos posibles
- 3.- Evaluamos los caminos en función de una serie de reglas.

4.- Reordenamos los caminos posibles

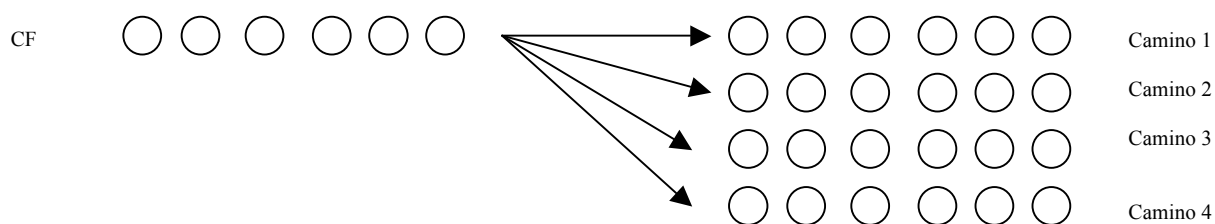


Fig.2

Hemos supuesto que el CF está en la voz de contralto y la nueva armonización en la voz de soprano. Por tanto, tomando la escala natural y en la primera especie del contrapunto severo, cada nota del CF puede ser armonizada de 13 maneras diferentes. Teniendo en cuenta que el CF tiene 6 notas, el número de caminos posible es de $13^6=4826809$ caminos, para lo que se necesita una potencia de cálculo importante.

Una primera solución llevada a cabo es aplicar de entrada la regla 1 del epígrafe 4.1 (*A cada nota del cantus firmus (CF) se le ha de añadir una nota consonante*). Sabiendo que las consonancias perfectas son unísono, tercera, quinta, sexta y octava, estas son las notas a añadir, de acuerdo con la tesitura. Para que haya posibilidad de cruzamiento de voces, añadimos también una tercera inferior. Para sobrepasar el ámbito de octava, añadimos una décima. De esta manera hemos simplificado el cálculo de manera considerable. El número de caminos ahora es de $6^6=46656$.

El procedimiento empleado consiste en descartar o penalizar caminos en función de las reglas del contrapunto severo.

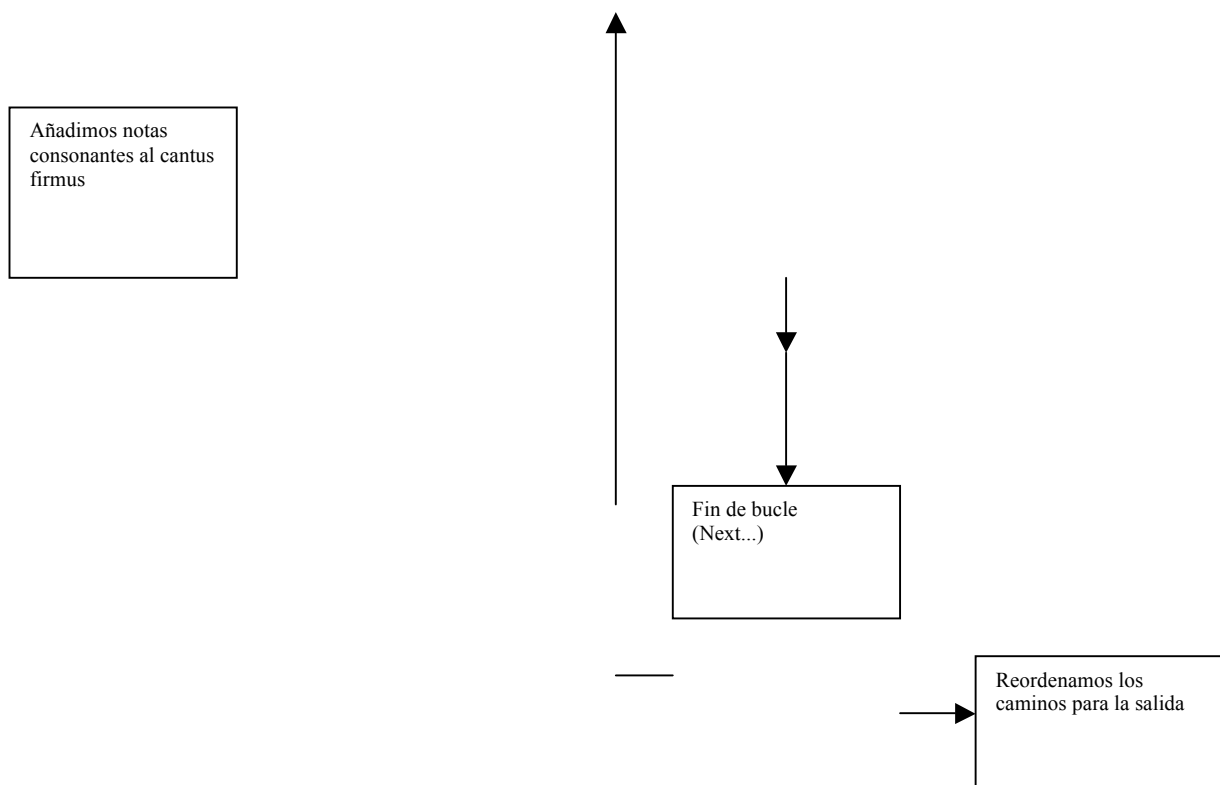
Las reglas que eliminan caminos son:

- 1.- Eliminar unísonos paralelos
- 2.- Eliminar quintas paralelas
- 3.- Eliminar octavas paralelas
- 4.- Eliminar octavas compuestas (unísono-octava, octava-unísono)
- 5.- No empezar ni acabar por sexta
- 6.- No permitir la quinta disminuida como intervalo armónico
- 7.- No permitir octavas resultantes, salvo las de trompa
- 8.- No permitir quintas resultantes, salvo las de trompa
- 9.- No saltos mayores que el de octava
- 10.- No saltos de séptima
- 11.- No dos saltos en la misma dirección.

Las reglas que penalizan caminos, con una penalización entre 0-1 son:

- 12.- Excesivos saltos
- 13.- No acabar con la tónica
- 14.- Repetir una misma nota
- 15.- Aparición de tritonos melódicos
- 16.- Aparición de 3 o más terceras seguidas
- 17.- Aparición de 3 o más sextas seguidas
- 18.- Número excesivo de unísonos.

Estableciendo un umbral de penalización, los caminos que no sobrepasen ese umbral serán eliminados. El procedimiento es sencillo, eficaz y rápido, a pesar de haber sido implementado en Visual Basic.



El algoritmo de programación es el siguiente:

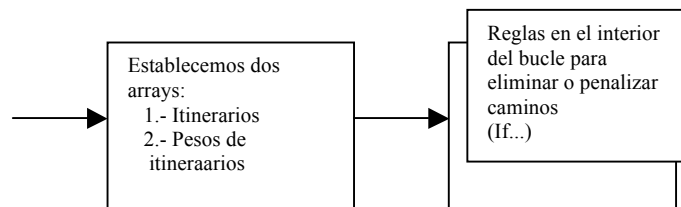


Fig. 3

6.2.- Segunda especie

En este caso tomamos como base parte de lo ya implementado en la primera especie. Teniendo en cuenta las reglas 12 y 13, en los tiempos fuertes de la segunda especie, siempre debe haber consonancia, por lo que el procedimiento empleado es

similar, con la salvedad de la modificación de algunas reglas. Hasta el momento, el esquema aplicado es el mismo que el de la figura 3.

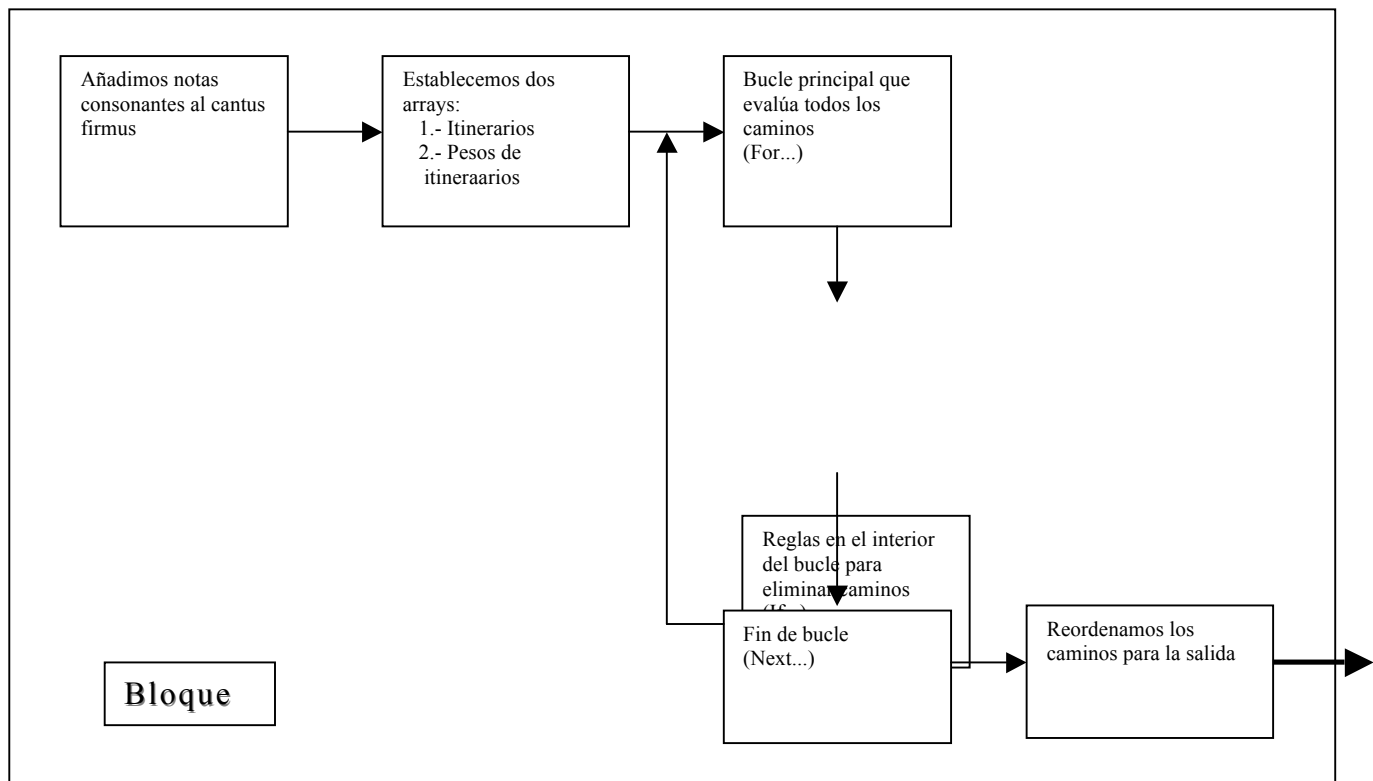
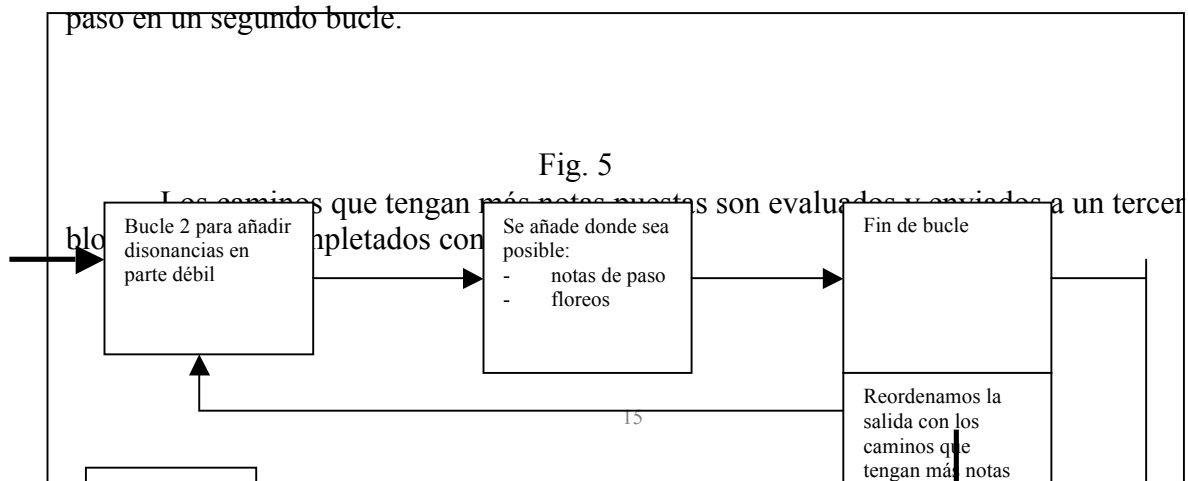


Fig. 4

Las reglas del interior del bucle 1 son:

- 1ª.- Eliminamos unísonos paralelos
- 2ª.- Eliminamos quintas paralelas
- 3ª.- Eliminamos octavas paralelas
- 4ª.- Eliminamos octavas compuestas (unísono-octava)
- 5ª.- No empezar por sexta ni acabar por sexta
- 6ª.- No quinta disminuida como intervalo armónico
- 7ª.- No saltos mayores que el de octava
- 8ª.- No saltos de 7ª

Una vez reordenados estos caminos, intentamos completarlos con floreos y notas de paso en un segundo bucle.



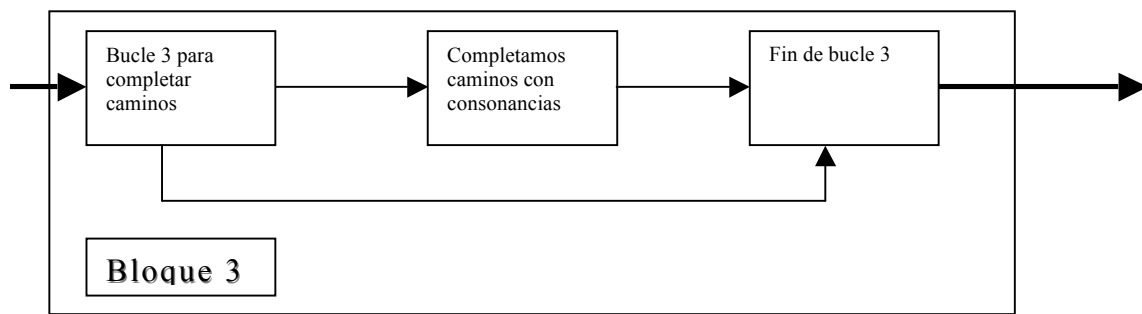


Fig.6

Estos caminos al final son evaluados con reglas para eliminar y penalizar caminos.

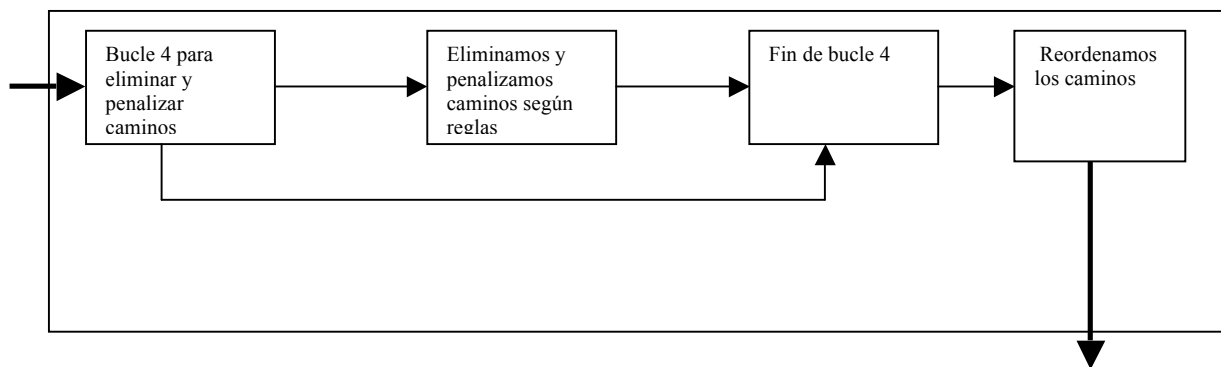


Fig. 7

7.- Implementación

A la hora de llevar a la práctica el sistema, hemos decidido hacerlo en lenguaje Visual Basic 6.0 por los siguientes motivos:

- 1.- Facilidad de manejar y trabajar con ficheros de los que importar y exportar datos de sonido (ficheros midi).
- 2.- Posibilidad de ampliación del trabajo de manera cómoda.
- 3.- Comodidad del entorno gráfico y facilidad para crear un entorno gráfico musical
- 4.- Posibilidad de mejorar algunas funciones por medio de ficheros *.dll en otros lenguajes más rápidos como el C.
- 5.- Facilidad y rapidez en la programación

Evidentemente este lenguaje tiene sus inconvenientes como:

- 1.- Relativa lentitud de ejecución del programa.
- 2.- Dificultad de la programación de reglas. Las reglas no son tan intuitivas como en otros entornos (NEXPERT o PROLOG, por ejemplo).

Para poder escuchar las armonizaciones nos hemos servido de un programa freeware en MS-Dos denominado “T2mf.exe”, con el que creamos nuestro fichero *.mid que será abierto por el programa “mplayer2.exe” de “windows”. Este programa ha sido creado por:

Piet van Oostrum, Dept of Computer Science, Utrecht University,

P.O. Box 80.089, 2508 TB Utrecht, The Netherlands

email: piet@cs.ruu.nl

8.- Manual de usuario

El funcionamiento del programa es bastante simple e intuitivo. Una vez instalado correctamente, se nos presenta en la pantalla principal un menú con tres opciones:

- Fichero: En este menú sólo tenemos la opción Salir para finalizar el programa
- Ayuda: Con la típica opción “Acerca de...” con los datos del programa
- Proceso: Esta opción es el corazón del programa. La explicamos a continuación con más detalle.

8.1.- Menú Proceso

Dentro de este menú, la opción **Crear CF** nos permite introducir un cantus firmus gráficamente desde la pantalla del ordenador. En la parte superior de la pantalla aparecen las notas de la escala natural en notación anglosajona, con los signos + y – para subir o bajar las notas, mientras que en la parte inferior aparece la notación tradicional de un cantus firmus para voz de contralto en clave de Do en tercera.

La opción play permite escuchar el cantus firmus. Este cantus firmus se escribe en un fichero midi denominado “Cf.mid” y en forma de texto en “Vdef.dat”. la manera de escribir las notas en el fichero de texto es mediante números que indiquen las notas separados por retornos de carro con la codificación midi. Por ejemplo:

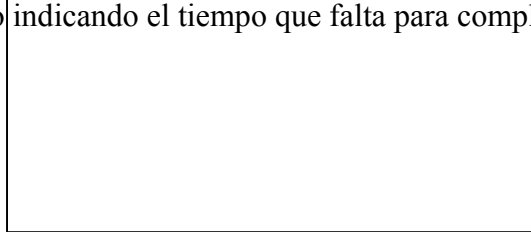
60	C	do
62	D	re
64	E	mi
65	F	fa
62	D	re
60	C	do

Fig. 8

Cuando pulsamos O.K., los valores escogidos se almacenan automáticamente en el fichero “Vdef.dat”, que será el fichero que se cargue al arrancar el programa.

La opción **Primera Especie** nos abre una pantalla con una serie de opciones. En la parte izquierda de la pantalla tenemos una serie de reglas en las que podemos seleccionar el grado de penalización que va a sufrir el camino correspondiente si incumple alguna de ellas. Una vez que pulsemos O.K., obtendremos todas las armonizaciones posibles que superen el límite que nosotros pongamos. La pantalla que se abre con la opción **Segunda Especie** es similar, excepto en el cambio de reglas.

En ambas pantallas, en el margen inferior izquierdo, aparece una barra de progreso indicando el tiempo que falta para completar el cálculo.



9.- Código fuente

1.-ContrapuntoSevero(Contrapunto Severo.Frm)

```
Private Sub AcercaDe_Click()
Load Acerca
Acerca.Show
End Sub

Private Sub FicheroSalir_Click()
End
End Sub

Private Sub Form_Load()

Open "Vdef.dat" For Input As #1
Dim Línea As String

For i = 0 To 5
Line Input #1, Línea
cantus_firmus(i) = Val(Línea)
```

```

Next i

Close #1

End Sub

Private Sub Procesol_Click()
Load PrimeraEsp
PrimeraEsp.Show
End Sub

Private Sub Proceso2_Click()
Load SegundaEsp
SegundaEsp.Show
End Sub

Private Sub ProcesoCF_Click()
Load CrearCF
CrearCF.Show
End Sub

```

2.-CrearCF(CrearCF.Frm)

```

Private Sub Command1_Click()
'Conversión Numérica

For i = 0 To 5
If Notas(i).Text = "e" Then cantus_firmus(i) = 52
If Notas(i).Text = "f" Then cantus_firmus(i) = 53
If Notas(i).Text = "g" Then cantus_firmus(i) = 55
If Notas(i).Text = "a" Then cantus_firmus(i) = 57
If Notas(i).Text = "b" Then cantus_firmus(i) = 59
If Notas(i).Text = "C" Then cantus_firmus(i) = 60
If Notas(i).Text = "D" Then cantus_firmus(i) = 62
If Notas(i).Text = "E" Then cantus_firmus(i) = 64
If Notas(i).Text = "F" Then cantus_firmus(i) = 65
If Notas(i).Text = "G" Then cantus_firmus(i) = 67
If Notas(i).Text = "A" Then cantus_firmus(i) = 69
Next i

'Fichero "CF.txt" para salida

Open "CF.txt" For Output As #1
Print #1, "MFile 0 1 24"
Print #1, "MTrk"
Print #1, "0 Tempo 300000"
Print #1, "0 On ch=1 n="; cantus_firmus(0); " v=64"
Print #1, "48 On ch=1 n="; cantus_firmus(0); " v=0"
Print #1, "96 On ch=1 n="; cantus_firmus(1); " v=64"
Print #1, "144 On ch=1 n="; cantus_firmus(1); " v=0"
Print #1, "192 On ch=1 n="; cantus_firmus(2); " v=64"
Print #1, "240 On ch=1 n="; cantus_firmus(2); " v=0"
Print #1, "288 On ch=1 n="; cantus_firmus(3); " v=64"
Print #1, "336 On ch=1 n="; cantus_firmus(3); " v=0"
Print #1, "384 On ch=1 n="; cantus_firmus(4); " v=64"
Print #1, "432 On ch=1 n="; cantus_firmus(4); " v=0"
Print #1, "480 On ch=1 n="; cantus_firmus(5); " v=64"
Print #1, "528 On ch=1 n="; cantus_firmus(5); " v=0"

Print #1, "576 Meta TrkEnd"

```

```

Print #1, "TrkEnd"
Close #1

'Fichero "CF.mid"

Dim hWndAct As Long, vr, hr
hWndAct = GetActiveWindow()
vr = Shell("t2mf -r CF.txt CF.mid")
Do While GetActiveWindow() = hWndAct
    vr = DoEvents()
Loop
hWndAct = GetForegroundWindow()
Do While IsWindow(hWndAct)
    vr = DoEvents()
Loop

hr = Shell("mplayer2 CF.mid")

End Sub

Private Sub Command3_Click()

    For i = 0 To 5
        If Notas(i).Text = "e" Then cantus_firmus(i) = 52
        If Notas(i).Text = "f" Then cantus_firmus(i) = 53
        If Notas(i).Text = "g" Then cantus_firmus(i) = 55
        If Notas(i).Text = "a" Then cantus_firmus(i) = 57
        If Notas(i).Text = "b" Then cantus_firmus(i) = 59
        If Notas(i).Text = "C" Then cantus_firmus(i) = 60
        If Notas(i).Text = "D" Then cantus_firmus(i) = 62
        If Notas(i).Text = "E" Then cantus_firmus(i) = 64
        If Notas(i).Text = "F" Then cantus_firmus(i) = 65
        If Notas(i).Text = "G" Then cantus_firmus(i) = 67
        If Notas(i).Text = "A" Then cantus_firmus(i) = 69
    Next i

    Open "Vdef.dat" For Output As #1
    For i = 0 To 5
        Print #1, cantus_firmus(i)
    Next i
    Close #1

    CrearCF.Hide
End Sub

Private Sub Form_Load()
    For i = 0 To 5
        If cantus_firmus(i) = 52 Then Notas(i).Text = "e"
        If cantus_firmus(i) = 53 Then Notas(i).Text = "f"
        If cantus_firmus(i) = 55 Then Notas(i).Text = "g"
        If cantus_firmus(i) = 57 Then Notas(i).Text = "a"
        If cantus_firmus(i) = 59 Then Notas(i).Text = "b"
        If cantus_firmus(i) = 60 Then Notas(i).Text = "C"
        If cantus_firmus(i) = 62 Then Notas(i).Text = "D"
        If cantus_firmus(i) = 64 Then Notas(i).Text = "E"
        If cantus_firmus(i) = 65 Then Notas(i).Text = "F"
        If cantus_firmus(i) = 67 Then Notas(i).Text = "G"
        If cantus_firmus(i) = 69 Then Notas(i).Text = "A"
    Next i

```

```

For i = 0 To 5
If Notas(i).Text = "e" Then Redonda(i).Move 1150 + (i * 700), 2415
If Notas(i).Text = "f" Then Redonda(i).Move 1150 + (i * 700), 2310
If Notas(i).Text = "g" Then Redonda(i).Move 1150 + (i * 700), 2200
If Notas(i).Text = "a" Then Redonda(i).Move 1150 + (i * 700), 2125
If Notas(i).Text = "b" Then Redonda(i).Move 1150 + (i * 700), 2050
If Notas(i).Text = "C" Then Redonda(i).Move 1150 + (i * 700), 1980
If Notas(i).Text = "D" Then Redonda(i).Move 1150 + (i * 700), 1890
If Notas(i).Text = "E" Then Redonda(i).Move 1150 + (i * 700), 1785
If Notas(i).Text = "F" Then Redonda(i).Move 1150 + (i * 700), 1680
If Notas(i).Text = "G" Then Redonda(i).Move 1150 + (i * 700), 1600
If Notas(i).Text = "A" Then Redonda(i).Move 1150 + (i * 700), 1500
Next i
End Sub

```

```

Private Sub Labell_Click()

```

```

For i = 0 To 5

```

```

If Notas(i).Text = "e" Then Redonda(i).Move 1150 + (i * 700), 2415
If Notas(i).Text = "f" Then Redonda(i).Move 1150 + (i * 700), 2310
If Notas(i).Text = "g" Then Redonda(i).Move 1150 + (i * 700), 2200
If Notas(i).Text = "a" Then Redonda(i).Move 1150 + (i * 700), 2125
If Notas(i).Text = "b" Then Redonda(i).Move 1150 + (i * 700), 2050
If Notas(i).Text = "C" Then Redonda(i).Move 1150 + (i * 700), 1980
If Notas(i).Text = "D" Then Redonda(i).Move 1150 + (i * 700), 1890
If Notas(i).Text = "E" Then Redonda(i).Move 1150 + (i * 700), 1785
If Notas(i).Text = "F" Then Redonda(i).Move 1150 + (i * 700), 1680
If Notas(i).Text = "G" Then Redonda(i).Move 1150 + (i * 700), 1600
If Notas(i).Text = "A" Then Redonda(i).Move 1150 + (i * 700), 1500
Next i
End Sub

```

```

Private Sub Text1_Change()

```

```

End Sub

```

```

Private Sub Notas_Change(Index As Integer)

```

```

For i = 0 To 5
If Notas(i).Text = "e" Then Redonda(i).Move 1150 + (i * 700), 2415
If Notas(i).Text = "f" Then Redonda(i).Move 1150 + (i * 700), 2310
If Notas(i).Text = "g" Then Redonda(i).Move 1150 + (i * 700), 2200
If Notas(i).Text = "a" Then Redonda(i).Move 1150 + (i * 700), 2125
If Notas(i).Text = "b" Then Redonda(i).Move 1150 + (i * 700), 2050
If Notas(i).Text = "C" Then Redonda(i).Move 1150 + (i * 700), 1980
If Notas(i).Text = "D" Then Redonda(i).Move 1150 + (i * 700), 1890
If Notas(i).Text = "E" Then Redonda(i).Move 1150 + (i * 700), 1785
If Notas(i).Text = "F" Then Redonda(i).Move 1150 + (i * 700), 1680
If Notas(i).Text = "G" Then Redonda(i).Move 1150 + (i * 700), 1600
If Notas(i).Text = "A" Then Redonda(i).Move 1150 + (i * 700), 1500
Next i
End Sub

```

```

Private Sub restar_Click(Index As Integer)

```

```

If Notas(Index).Text = "e" Then Notas(Index).Text = "e"
If Notas(Index).Text = "f" Then Notas(Index).Text = "e"
If Notas(Index).Text = "g" Then Notas(Index).Text = "f"
If Notas(Index).Text = "a" Then Notas(Index).Text = "g"
If Notas(Index).Text = "b" Then Notas(Index).Text = "a"
If Notas(Index).Text = "C" Then Notas(Index).Text = "b"

```

```

If Notas(Index).Text = "D" Then Notas(Index).Text = "C"
If Notas(Index).Text = "E" Then Notas(Index).Text = "D"
If Notas(Index).Text = "F" Then Notas(Index).Text = "E"
If Notas(Index).Text = "G" Then Notas(Index).Text = "F"
If Notas(Index).Text = "A" Then Notas(Index).Text = "G"

```

```
End Sub
```

```
Private Sub sumar_Click(Index As Integer)
```

```

If Notas(Index).Text = "A" Then Notas(Index).Text = "A"
If Notas(Index).Text = "G" Then Notas(Index).Text = "A"
If Notas(Index).Text = "F" Then Notas(Index).Text = "G"
If Notas(Index).Text = "E" Then Notas(Index).Text = "F"
If Notas(Index).Text = "D" Then Notas(Index).Text = "E"
If Notas(Index).Text = "C" Then Notas(Index).Text = "D"
If Notas(Index).Text = "b" Then Notas(Index).Text = "C"
If Notas(Index).Text = "a" Then Notas(Index).Text = "b"
If Notas(Index).Text = "g" Then Notas(Index).Text = "a"
If Notas(Index).Text = "f" Then Notas(Index).Text = "g"
If Notas(Index).Text = "e" Then Notas(Index).Text = "f"

```

```
End Sub
```

3.-PrimeraEsp(PrimeraEsp.Frm)

```
Private Sub Command1_Click()
```

```
Picture1.Cls
```

```

'Array de valores posibles del cantus firmus
Dim notas_posib(20) As Integer

```

```

notas_posib(0) = (52) 'e
notas_posib(1) = (53) 'f
notas_posib(2) = (55) 'g
notas_posib(3) = (57) 'a
notas_posib(4) = (59) 'b
notas_posib(5) = (60) 'C
notas_posib(6) = (62) 'D
notas_posib(7) = (64) 'E
notas_posib(8) = (65) 'F
notas_posib(9) = (67) 'G
notas_posib(10) = (69) 'A
notas_posib(11) = (71) 'B
notas_posib(12) = (72) 'C1
notas_posib(13) = (74) 'D1
notas_posib(14) = (76) 'E1
notas_posib(15) = (77) 'F1
notas_posib(16) = (79) 'G1
notas_posib(17) = (81) 'A1
notas_posib(18) = (83) 'B1
notas_posib(19) = (84) 'C2

```

```

' /*****
' /* Aplicación de reglas*/
' /*****

```

```
Dim nodos_notas(6, 7) As Integer
```

```

' /*****
*** /
' /* 1º regla: A cada nota del CF se le ha de añadir una nota
consonante */
' /*****
*** /

p = 0
For i = 0 To 5

    For n = 0 To 10

        If cantus_firmus(i) = notas_posib(n) Then
            nodos_notas(i, p) = notas_posib(n - 2) '/* tercera inferior */
            If nodos_notas(i, p) >= 60 Then p = p + 1
            If nodos_notas(i, p) > 81 Then nodos_notas(i, p) =
nodos_notas(i, p - 1)
            If p >= 6 Then
                n = 10
                GoTo Etiqueta
            End If
            nodos_notas(i, p) = notas_posib(n + 0) '/* unísono */
            If nodos_notas(i, p) >= 60 Then p = p + 1
            If nodos_notas(i, p) > 81 Then nodos_notas(i, p) =
nodos_notas(i, p - 1)
            If p >= 6 Then
                n = 10
                GoTo Etiqueta
            End If
            nodos_notas(i, p) = notas_posib(n + 2) ' /* tercera */
            If nodos_notas(i, p) >= 60 Then p = p + 1
            If nodos_notas(i, p) > 81 Then nodos_notas(i, p) =
nodos_notas(i, p - 1)
            If p >= 6 Then
                n = 10
                GoTo Etiqueta
            End If
            nodos_notas(i, p) = notas_posib(n + 4) ' /* quinta */
            If nodos_notas(i, p) >= 60 Then p = p + 1
            If nodos_notas(i, p) > 81 Then nodos_notas(i, p) =
nodos_notas(i, p - 1)
            If p >= 6 Then
                n = 10
                GoTo Etiqueta
            End If
            nodos_notas(i, p) = notas_posib(n + 5) '/* sexta */
            If nodos_notas(i, p) >= 60 Then p = p + 1
            If nodos_notas(i, p) > 81 Then nodos_notas(i, p) =
nodos_notas(i, p - 1)
            If p >= 6 Then
                n = 10
                GoTo Etiqueta
            End If
            nodos_notas(i, p) = notas_posib(n + 7) '/* octava */
            If nodos_notas(i, p) >= 60 Then p = p + 1
            If nodos_notas(i, p) > 81 Then nodos_notas(i, p) =
nodos_notas(i, p - 1)
            If p >= 6 Then
                n = 10
                GoTo Etiqueta
            End If
            nodos_notas(i, p) = notas_posib(n + 9) '/* décima */
            If nodos_notas(i, p) >= 60 Then p = p + 1

```

```

        If nodos_notas(i, p) > 81 Then nodos_notas(i, p) =
nodos_notas(i, p - 1)
        If p >= 6 Then
            n = 10
            GoTo Etiqueta
        End If

Etiqueta:    End If
            p = 0

        Next n

    Next i

```

```

' /*****
' /* Búsqueda de los mejores itinerarios */
' *****/

'static char peso_itinerario[1000];
'static char peso_itinerario2[1000];

' /*****
' /* Supongamos que el cantus firmus tiene 6 notas */
' *****/

'Dim itinerario(50000, 5) As Integer
Dim peso_itinerario(50000) As Single

Dim a(10) As Integer
n_pas = 0

b = 0
    For a0 = 0 To 5
        For a1 = 0 To 5
            For a2 = 0 To 5
                For a3 = 0 To 5
                    For a4 = 0 To 5
                        For a5 = 0 To 5

```

```

                            itinerario(b, 0) = nodos_notas(0, a0)
                            itinerario(b, 1) = nodos_notas(1, a1)
                            itinerario(b, 2) = nodos_notas(2, a2)
                            itinerario(b, 3) = nodos_notas(3, a3)
                            itinerario(b, 4) = nodos_notas(4, a4)
                            itinerario(b, 5) = nodos_notas(5, a5)

```

```

                            b = b + 1

```

```

                    Next a5
                Next a4
            Next a3

```

```

Next a2
Next a1
'Print a0, b
Next a0

camino_total = b

' /*****
' /*****
' /* REGLAS DE CONDUCCIÓN DE VOCES */
' /*****
' /*****

ProgressBar1.Max = camino_total
For i = 0 To camino_total

    ProgressBar1.Value = i
    For b = 0 To 4

' /*****
' /* REGLAS PARA ELIMINAR CAMINOS */
' /*****

' /*** 1ª.- Eliminamos unísonos paralelos ***/

    If itinerario(i, b) - cantus_firmus(b) = 0 Then      ' /* caminos
con unísonos */

        If itinerario(i, b + 1) - cantus_firmus(b + 1) = 0 Then

            peso_itinerario(i) = peso_itinerario(i) + 1

        End If
    End If

' /*** 2ª.- Eliminamos quintas paralelas ***/

    If itinerario(i, b) - cantus_firmus(b) = 7 Then      ' /* caminos
con 5ªs */

        If itinerario(i, b + 1) - cantus_firmus(b + 1) = 7 Then

            peso_itinerario(i) = peso_itinerario(i) + 1

        End If
    End If

' /*** 3ª.- Eliminamos octavas paralelas ***/

    If itinerario(i, b) - cantus_firmus(b) = 12 Then ' /* caminos con
octavas */

        If itinerario(i, b + 1) - cantus_firmus(b + 1) = 12 Then

```

```

    peso_itinerario(i) = peso_itinerario(i) + 1

    End If
    End If

' /*** 4ª.- Eliminamos octavas compuestas (unísono-octava) *** /

If itinerario(i, b) - cantus_firmus(b) = 0 Then

    If itinerario(i, b + 1) - cantus_firmus(b + 1) = 12 Then

        peso_itinerario(i) = peso_itinerario(i) + 1

        End If
        End If

' /*** 4bª.- Eliminamos octavas compuestas (octava-unísono) *** /

If itinerario(i, b) - cantus_firmus(b) = 12 Then

    If itinerario(i, b + 1) - cantus_firmus(b + 1) = 0 Then

        peso_itinerario(i) = peso_itinerario(i) + 1

        End If
        End If

' /*** 5ª.- No empezar por sexta *** /

If itinerario(i, 0) - cantus_firmus(0) = 9 Then

    peso_itinerario(i) = peso_itinerario(i) + 1

    End If

' /*** 5bª.- No acabar por sexta *** /

If itinerario(i, 5) - cantus_firmus(5) = 9 Then

    peso_itinerario(i) = peso_itinerario(i) + 1

    End If

' /*** 6ª.- No quinta disminuida *** /

If itinerario(i, b) - cantus_firmus(b) = 6 Then

    ' peso_itinerario(i) = peso_itinerario(i) + 1

    End If

' /*** 7ª.- No octavas resultantes salvo las de trompa *** /

    If itinerario(i, b + 1) - cantus_firmus(b + 1) = 12 Then
    If itinerario(i, b + 1) - itinerario(i, b) > 2 Then
    If cantus_firmus(b + 1) - cantus_firmus(b) > 0 Then
    peso_itinerario(i) = peso_itinerario(i) + 1
    End If
    End If
    End If

    If itinerario(i, b + 1) - cantus_firmus(b + 1) = 12 Then
    If itinerario(i, b) - itinerario(i, b + 1) > 2 Then

```

```

If cantus_firmus(b) - cantus_firmus(b + 1) > 0 Then
peso_itinerario(i) = peso_itinerario(i) + 1
End If
End If
End If

' /*** 8ª.- No quintas resultantes salvo las de trompa ***/

If itinerario(i, b + 1) - cantus_firmus(b + 1) = 7 Then
If itinerario(i, b + 1) - itinerario(i, b) > 2 Then
If cantus_firmus(b + 1) - cantus_firmus(b) > 0 Then
peso_itinerario(i) = peso_itinerario(i) + 1
End If
End If
End If

If itinerario(i, b + 1) - cantus_firmus(b + 1) = 7 Then
If itinerario(i, b) - itinerario(i, b + 1) > 2 Then
If cantus_firmus(b) - cantus_firmus(b + 1) > 0 Then
peso_itinerario(i) = peso_itinerario(i) + 1
End If
End If
End If

' /*** 9ª.- No saltos mayores que el de octava ***/

If Abs(itinerario(i, b + 1) - itinerario(i, b)) > 12 Then
peso_itinerario(i) = peso_itinerario(i) + 1
End If

' /*** 10ª.- No saltos de 7ª ***/
If Abs(itinerario(i, b + 1) - itinerario(i, b)) = 11 Then
peso_itinerario(i) = peso_itinerario(i) + 1
End If

If Abs(itinerario(i, b + 1) - itinerario(i, b)) = 10 Then
peso_itinerario(i) = peso_itinerario(i) + 1
End If

' /*** 11ª.- No dos saltos en la misma dirección ***/
If itinerario(i, b) - itinerario(i, b + 1) > 2 Then
If itinerario(i, b + 1) - itinerario(i, b + 2) > 2 Then
peso_itinerario(i) = peso_itinerario(i) + 1
End If
End If

If itinerario(i, b) - itinerario(i, b + 1) < 2 Then
If itinerario(i, b + 1) - itinerario(i, b + 2) < 2 Then
peso_itinerario(i) = peso_itinerario(i) + 1
End If
End If

' /*****/
' /* REGLAS PARA PENALIZAR CAMINOS */
' /*****/

' /*** 7ª.- Sólo grado conjunto ***/

If (itinerario(i, b + 1) - itinerario(i, b)) > 2 Then

```

```

    peso_itinerario(i) = peso_itinerario(i) + (GradoConjunto.Value
/ 100)

    End If

    If (itinerario(i, b) - itinerario(i, b + 1)) > 2 Then

        peso_itinerario(i) = peso_itinerario(i) + (GradoConjunto.Value
/ 100)

        End If

'/** 8ª.- Acabar con la tónica ***/

    If itinerario(i, 5) - cantus_firmus(5) <> 12 Then

    If itinerario(i, 5) - cantus_firmus(5) <> 0 Then

        peso_itinerario(i) = peso_itinerario(i) + (TónicaFin.Value /
100)

        End If
        End If

'/** 9ª.- Nota repetida ***/

    If itinerario(i, b) = itinerario(i, b + 1) Then

        peso_itinerario(i) = peso_itinerario(i) +
(NotaRepetida.Value / 100)

        End If

'/** 10ª.- Tritonos melódicos ***/

    If Abs(itinerario(i, b) - itinerario(i, b + 1)) = 6 Then
        peso_itinerario(i) = peso_itinerario(i) + (TritonoM.Value /
100)
    End If

'/** 11ª.- Terceras seguidas ***/

    If Abs(itinerario(i, b) - cantus_firmus(b)) = 4 Or
Abs(itinerario(i, b) - cantus_firmus(b)) = 3 Then
        If Abs(itinerario(i, b + 1) - cantus_firmus(b + 1)) =
4 Or Abs(itinerario(i, b + 1) - cantus_firmus(b + 1)) = 3 Then
            If Abs(itinerario(i, b + 2) - cantus_firmus(b + 2)) =
4 Or Abs(itinerario(i, b + 2) - cantus_firmus(b + 2)) = 3 Then
                peso_itinerario(i) = peso_itinerario(i) +
(TercerasSeg.Value / 100)
            End If
            End If
            End If

'/** 11b.- Sextas seguidas ***/

    If Abs(itinerario(i, b) - cantus_firmus(b)) = 8 Or
Abs(itinerario(i, b) - cantus_firmus(b)) = 9 Then
        If Abs(itinerario(i, b + 1) - cantus_firmus(b + 1)) =
8 Or Abs(itinerario(i, b + 1) - cantus_firmus(b + 1)) = 9 Then
            If Abs(itinerario(i, b + 2) - cantus_firmus(b + 2)) =
8 Or Abs(itinerario(i, b + 2) - cantus_firmus(b + 2)) = 9 Then
                peso_itinerario(i) = peso_itinerario(i) +
(TercerasSeg.Value / 100)
            End If
        End If
    End If

```

```

        End If
        End If
        End If

'/** 12.- Excesivos unísonos **/
    If itinerario(i, b) - cantus_firmus(b) = 0 Then
        peso_itinerario(i) = peso_itinerario(i) + (Unís.Value /
100)
        End If

        If itinerario(i, b + 1) - cantus_firmus(b + 1) = 0 Then
        peso_itinerario(i) = peso_itinerario(i) + (Unís.Value /
100)
        End If

    Next b
    Next i

' /***** Reordenamos itinerarios *****/
' /***** Reordenamos itinerarios *****/
' /***** Reordenamos itinerarios *****/

    m = 0

    For i = 0 To camino_total

        If peso_itinerario(i) < (Límite.Value / 10) Then

            For b = 0 To 5

                itinerario(m, b) = itinerario(i, b)

                Picture1.Print itinerario(m, b); " ";
                Next b
                Picture1.Print i

                peso_itinerario(m) = peso_itinerario(i)

                m = m + 1
            End If

        Next i

    caminos = m
    Text10.Text = caminos

    If caminos > 0 Then
        Unload Pentagramas
        Load Pentagramas
        Pentagramas.Show
    End If

End Sub

Private Sub Command2_Click()

```

```

PrimeraEsp.Hide
End Sub

Private Sub GradoConjunto_Change()
Text7.Text = Format(GradoConjunto.Value / 100, "0.00")
End Sub

Private Sub GradoConjunto_Scroll()
Text7.Text = Format(GradoConjunto.Value / 100, "0.00")
End Sub

Private Sub Límite_Change()
Lim.Text = Format((Límite.Value / 10), "0.00")

End Sub

Private Sub NotaRepetida_Change()
Text9.Text = Format(NotaRepetida.Value / 100, "0.00")
End Sub

Private Sub NotaRepetida_Scroll()
Text9.Text = Format(NotaRepetida.Value / 100, "0.00")
End Sub

Private Sub TercerasSeg_Change()
Text2.Text = Format(TercerasSeg.Value / 100, "0.00")
End Sub

Private Sub TercerasSeg_Scroll()
Text2.Text = Format(TercerasSeg.Value / 100, "0.00")
End Sub

Private Sub TónicaFin_Change()
Text8.Text = Format(TónicaFin.Value / 100, "0.00")
End Sub

Private Sub TónicaFin_Scroll()
Text8.Text = Format(TónicaFin.Value / 100, "0.00")
End Sub

Private Sub TritonoM_Change()
Text1.Text = Format(TritonoM.Value / 100, "0.00")
End Sub

Private Sub TritonoM_Scroll()
Text1.Text = Format(TritonoM.Value / 100, "0.00")
End Sub

Private Sub Unís_Change()
Text3.Text = Format(Unís.Value / 100, "0.00")
End Sub

Private Sub Unís_Scroll()
Text3.Text = Format(Unís.Value / 100, "0.00")
End Sub

```

4.-Pentagramas(Pentagramas.Frm)

```

Private Sub Armon_Change()
Line8(0).Visible = False
Line8(1).Visible = False
Line8(2).Visible = False
Line8(3).Visible = False

pk = 0
For i = 0 To 5
p = 0
If Module1.itinerario(Armon.Text, i) = 60 Then

    Redonda(i + 6).Visible = True

    Redonda(i + 6).Move 1150 + (i * 700), 900
    Line8(pk).Visible = True
    Line8(pk).X1 = 1050 + (i * 700)
    Line8(pk).X2 = 1450 + (i * 700)
    Line8(pk).Y1 = 1890
    Line8(pk).Y2 = 1890

    p = 1
    pk = pk + 1
End If
If Module1.itinerario(Armon.Text, i) = 62 Then Redonda(i + 6).Move
1150 + (i * 700), 840
If Module1.itinerario(Armon.Text, i) = 64 Then Redonda(i + 6).Move
1150 + (i * 700), 735
If Module1.itinerario(Armon.Text, i) = 65 Then Redonda(i + 6).Move
1150 + (i * 700), 630
If Module1.itinerario(Armon.Text, i) = 67 Then Redonda(i + 6).Move
1150 + (i * 700), 535
If Module1.itinerario(Armon.Text, i) = 69 Then Redonda(i + 6).Move
1150 + (i * 700), 460
If Module1.itinerario(Armon.Text, i) = 71 Then Redonda(i + 6).Move
1150 + (i * 700), 410
If Module1.itinerario(Armon.Text, i) = 72 Then Redonda(i + 6).Move
1150 + (i * 700), 315
If Module1.itinerario(Armon.Text, i) = 74 Then Redonda(i + 6).Move
1150 + (i * 700), 210
If Module1.itinerario(Armon.Text, i) = 76 Then Redonda(i + 6).Move
1150 + (i * 700), 115
If Module1.itinerario(Armon.Text, i) = 77 Then Redonda(i + 6).Move
1150 + (i * 700), 10
If Module1.itinerario(Armon.Text, i) = 79 Then Redonda(i + 6).Move
1150 + (i * 700), -80

Next i
End Sub

Private Sub Command1_Click()
Pentagramas.Hide
End Sub

Private Sub Form_Load()
'Tabla de conversión para cantus firmus

For i = 0 To 5

If Module1.cantus_firmus(i) = 53 Then Redonda(i).Move 1150 + (i *
700), 2310
If Module1.cantus_firmus(i) = 55 Then Redonda(i).Move 1150 + (i *
700), 2200

```

```

If Module1.cantus_firmus(i) = 57 Then Redonda(i).Move 1150 + (i * 700), 2125
If Module1.cantus_firmus(i) = 59 Then Redonda(i).Move 1150 + (i * 700), 2050
If Module1.cantus_firmus(i) = 60 Then Redonda(i).Move 1150 + (i * 700), 1980
If Module1.cantus_firmus(i) = 62 Then Redonda(i).Move 1150 + (i * 700), 1890
If Module1.cantus_firmus(i) = 64 Then Redonda(i).Move 1150 + (i * 700), 1785
If Module1.cantus_firmus(i) = 65 Then Redonda(i).Move 1150 + (i * 700), 1680
If Module1.cantus_firmus(i) = 67 Then Redonda(i).Move 1150 + (i * 700), 1600
If Module1.cantus_firmus(i) = 69 Then Redonda(i).Move 1150 + (i * 700), 1500
Next i

'Tabla de conversión para resultado
pk = 0
For i = 0 To 5
If Module1.itinerario(Armon.Text, i) = 60 Then
    Redonda(i + 6).Visible = True

    Redonda(i + 6).Move 1150 + (i * 700), 900
    Line8(pk).Visible = True
    Line8(pk).X1 = 1050 + (i * 700)
    Line8(pk).X2 = 1450 + (i * 700)
    Line8(pk).Y1 = 1890
    Line8(pk).Y2 = 1890
    pk = pk + 1
    p = 1
End If
If Module1.itinerario(0, i) = 62 Then Redonda(i + 6).Move 1150 + (i * 700), 840
If Module1.itinerario(0, i) = 64 Then Redonda(i + 6).Move 1150 + (i * 700), 735
If Module1.itinerario(0, i) = 65 Then Redonda(i + 6).Move 1150 + (i * 700), 630
If Module1.itinerario(0, i) = 67 Then Redonda(i + 6).Move 1150 + (i * 700), 535
If Module1.itinerario(0, i) = 69 Then Redonda(i + 6).Move 1150 + (i * 700), 460
If Module1.itinerario(0, i) = 71 Then Redonda(i + 6).Move 1150 + (i * 700), 410
If Module1.itinerario(0, i) = 72 Then Redonda(i + 6).Move 1150 + (i * 700), 315
If Module1.itinerario(0, i) = 74 Then Redonda(i + 6).Move 1150 + (i * 700), 210
If Module1.itinerario(0, i) = 76 Then Redonda(i + 6).Move 1150 + (i * 700), 115
If Module1.itinerario(0, i) = 77 Then Redonda(i + 6).Move 1150 + (i * 700), 10
If Module1.itinerario(0, i) = 79 Then Redonda(i + 6).Move 1150 + (i * 700), -80

Next i

End Sub

Private Sub HScroll11_Change()
Armon.Text = HScroll11.Value
If HScroll11.Value > caminos - 1 Then Armon.Text = caminos - 1

```

```

HScroll1.Max = caminos - 1
Picture1.Cls
For i = 0 To 5
Picture1.Print itinerario(HScroll1.Value, i); " ";
Next i
End Sub

Private Sub Play_Click()
Open "datos.txt" For Output As #1
Print #1, "MFile 0 1 24"
Print #1, "MTrk"
Print #1, "0 Tempo 300000"
Print #1, "0 On ch=1 n="; Module1.itinerario(Armon.Text, 0); " v=64"
Print #1, "0 On ch=2 n="; Module1.cantus_firmus(0); " v=64"
Print #1, "96 On ch=1 n="; Module1.itinerario(Armon.Text, 0); " v=0"
Print #1, "96 On ch=2 n="; Module1.cantus_firmus(0); " v=0"
Print #1, "96 On ch=1 n="; Module1.itinerario(Armon.Text, 1); " v=64"
Print #1, "96 On ch=2 n="; Module1.cantus_firmus(1); " v=64"
Print #1, "192 On ch=1 n="; Module1.itinerario(Armon.Text, 1); " v=0"
Print #1, "192 On ch=2 n="; Module1.cantus_firmus(1); " v=0"
Print #1, "192 On ch=1 n="; Module1.itinerario(Armon.Text, 2); " v=64"
Print #1, "192 On ch=2 n="; Module1.cantus_firmus(2); " v=64"
Print #1, "288 On ch=1 n="; Module1.itinerario(Armon.Text, 2); " v=0"
Print #1, "288 On ch=2 n="; Module1.cantus_firmus(2); " v=0"
Print #1, "288 On ch=1 n="; Module1.itinerario(Armon.Text, 3); " v=64"
Print #1, "288 On ch=2 n="; Module1.cantus_firmus(3); " v=64"
Print #1, "384 On ch=1 n="; Module1.itinerario(Armon.Text, 3); " v=0"
Print #1, "384 On ch=2 n="; Module1.cantus_firmus(3); " v=0"
Print #1, "384 On ch=1 n="; Module1.itinerario(Armon.Text, 4); " v=64"
Print #1, "384 On ch=2 n="; Module1.cantus_firmus(4); " v=64"
Print #1, "480 On ch=1 n="; Module1.itinerario(Armon.Text, 4); " v=0"
Print #1, "480 On ch=2 n="; Module1.cantus_firmus(4); " v=0"
Print #1, "480 On ch=1 n="; Module1.itinerario(Armon.Text, 5); " v=64"
Print #1, "480 On ch=2 n="; Module1.cantus_firmus(5); " v=64"
Print #1, "576 On ch=1 n="; Module1.itinerario(Armon.Text, 5); " v=0"
Print #1, "576 On ch=2 n="; Module1.cantus_firmus(5); " v=0"

Print #1, "576 Meta TrkEnd"
Print #1, "TrkEnd"
Close #1

'Ejecutamos T2mf

Dim hWndAct As Long, vr, hr

hWndAct = GetActiveWindow()
vr = Shell("t2mf -r datos.txt v2.mid")
Do While GetActiveWindow() = hWndAct
vr = DoEvents()
Loop
hWndAct = GetForegroundWindow()
Do While IsWindow(hWndAct)
vr = DoEvents()
Loop

hr = Shell("mplayer2 v2.mid")

End Sub

Private Sub Text1_Change()

```

```
End Sub
```

5.-SegundaEsp(SegundaEsp.Frm)

```
Private Sub Cadencia_Change()  
Text8.Text = Format(Cadencia.Value / 100, "0.00")  
End Sub
```

```
Private Sub Command1_Click()
```

```
    'Picture1.Cls
```

```
    'Array de valores posibles del cantus firmus  
    Dim notas_posib(20) As Integer
```

```
    notas_posib(0) = (52) 'e  
    notas_posib(1) = (53) 'f  
    notas_posib(2) = (55) 'g  
    notas_posib(3) = (57) 'a  
    notas_posib(4) = (59) 'b  
    notas_posib(5) = (60) 'C  
    notas_posib(6) = (62) 'D  
    notas_posib(7) = (64) 'E  
    notas_posib(8) = (65) 'F  
    notas_posib(9) = (67) 'G  
    notas_posib(10) = (69) 'A  
    notas_posib(11) = (71) 'B  
    notas_posib(12) = (72) 'C1  
    notas_posib(13) = (74) 'D1  
    notas_posib(14) = (76) 'E1  
    notas_posib(15) = (77) 'F1  
    notas_posib(16) = (79) 'G1  
    notas_posib(17) = (81) 'A1  
    notas_posib(18) = (83) 'B1  
    notas_posib(19) = (84) 'C2
```

```
    '/* ****/  
    '/* Aplicación de reglas*/  
    '/* ****/
```

```
        Dim nodos_notas(6, 7) As Integer
```

```
    '/* ****/  
    '/* 1º regla: A cada nota del CF se le ha de añadir una nota  
    consonante */  
    '/* ****/  
    '/* ****/
```

```
    p = 0  
    For i = 0 To 5
```

```
        For n = 0 To 10
```

```
            If cantus_firmus(i) = notas_posib(n) Then  
                If n > 2 Then  
                    nodos_notas(i, p) = notas_posib(n - 2) '/* tercera inferior */  
                    If nodos_notas(i, p) >= 60 Then p = p + 1  
                    If nodos_notas(i, p) > 81 Then nodos_notas(i, p) =  
nodos_notas(i, p - 1)  
                    If p >= 6 Then  
                        n = 10
```

```

        GoTo Etiqueta
    End If
End If
nodos_notas(i, p) = notas_posib(n + 0) '/* unísono */
    If nodos_notas(i, p) >= 60 Then p = p + 1
    If nodos_notas(i, p) > 81 Then nodos_notas(i, p) =
nodos_notas(i, p - 1)
    If p >= 6 Then
        n = 10
        GoTo Etiqueta
    End If
    nodos_notas(i, p) = notas_posib(n + 2) '/* tercera */
    If nodos_notas(i, p) >= 60 Then p = p + 1
    If nodos_notas(i, p) > 81 Then nodos_notas(i, p) =
nodos_notas(i, p - 1)
    If p >= 6 Then
        n = 10
        GoTo Etiqueta
    End If
    nodos_notas(i, p) = notas_posib(n + 4) '/* quinta */
    If nodos_notas(i, p) >= 60 Then p = p + 1
    If nodos_notas(i, p) > 81 Then nodos_notas(i, p) =
nodos_notas(i, p - 1)
    If p >= 6 Then
        n = 10
        GoTo Etiqueta
    End If
    nodos_notas(i, p) = notas_posib(n + 5) '/* sexta */
    If nodos_notas(i, p) >= 60 Then p = p + 1
    If nodos_notas(i, p) > 81 Then nodos_notas(i, p) =
nodos_notas(i, p - 1)
    If p >= 6 Then
        n = 10
        GoTo Etiqueta
    End If
    nodos_notas(i, p) = notas_posib(n + 7) '/* octava */
    If nodos_notas(i, p) >= 60 Then p = p + 1
    If nodos_notas(i, p) > 81 Then nodos_notas(i, p) =
nodos_notas(i, p - 1)
    If p >= 6 Then
        n = 10
        GoTo Etiqueta
    End If
    nodos_notas(i, p) = notas_posib(n + 9) '/* décima */
    If nodos_notas(i, p) >= 60 Then p = p + 1
    If nodos_notas(i, p) > 81 Then nodos_notas(i, p) =
nodos_notas(i, p - 1)
    If p >= 6 Then
        n = 10
        GoTo Etiqueta
    End If

Etiqueta:    End If
    p = 0

Next n

Next i

'/* ***** */
'/* Búsqueda de los mejores itinerarios */

```

```

' /*****/

'static char peso_itinerario[1000];
'static char peso_itinerario2[1000];

' /*****/
'/* Supongamos que el cantus firmus tiene 6 notas */
' /*****/

'Dim itinerario(50000, 5) As Integer
Dim peso_itinerario(50000) As Single

Dim a(10) As Integer
n_pas = 0

b = 0
  For a0 = 0 To 5
    For a1 = 0 To 5
      For a2 = 0 To 5
        For a3 = 0 To 5
          For a4 = 0 To 5
            For a5 = 0 To 5

              itinerario(b, 0) = nodos_notas(0, a0)
              itinerario(b, 1) = nodos_notas(1, a1)
              itinerario(b, 2) = nodos_notas(2, a2)
              itinerario(b, 3) = nodos_notas(3, a3)
              itinerario(b, 4) = nodos_notas(4, a4)
              itinerario(b, 5) = nodos_notas(5, a5)

              b = b + 1

            Next a5
          Next a4
        Next a3
      Next a2
    Next a1
    'Print a0, b
  Next a0

  camino_total = b

' /*****/
' /*****/
' /* REGLAS DE CONDUCCIÓN DE VOCES */
' /*****/
' /*****/

ProgressBar1.Max = camino_total
  For i = 0 To camino_total
    ProgressBar1.Value = i
    For b = 0 To 4

```

```

' /***** REGLAS PARA ELIMINAR CAMINOS */
' /***** 0ª.- Eliminamos caminos vacíos *****/

    If itinerario(i, b) = 0 Then peso_itinerario(i) =
peso_itinerario(i) + 10

' /**** 1ª.- Eliminamos unísonos paralelos ****/

    If itinerario(i, b) - cantus_firmus(b) = 0 Then      ' /* caminos
con unísonos */

        If itinerario(i, b + 1) - cantus_firmus(b + 1) = 0 Then

            peso_itinerario(i) = peso_itinerario(i) + 10

        End If
    End If

' /**** 2ª.- Eliminamos quintas paralelas ****/

    If itinerario(i, b) - cantus_firmus(b) = 7 Then      ' /* caminos
con 5ªs */

        If itinerario(i, b + 1) - cantus_firmus(b + 1) = 7 Then

            peso_itinerario(i) = peso_itinerario(i) + 10

        End If
    End If

' /**** 3ª.- Eliminamos octavas paralelas ****/

    If itinerario(i, b) - cantus_firmus(b) = 12 Then ' /* caminos con
octavas */

        If itinerario(i, b + 1) - cantus_firmus(b + 1) = 12 Then

            peso_itinerario(i) = peso_itinerario(i) + 10

        End If
    End If

' /**** 4ª.- Eliminamos octavas compuestas (unísono-octava) ****/

    If itinerario(i, b) - cantus_firmus(b) = 0 Then

        If itinerario(i, b + 1) - cantus_firmus(b + 1) = 12 Then

            peso_itinerario(i) = peso_itinerario(i) + 10

        End If
    End If

' /**** 4bª.- Eliminamos octavas compuestas (octava-unísono) ****/

```

```

If itinerario(i, b) - cantus_firmus(b) = 12 Then
    If itinerario(i, b + 1) - cantus_firmus(b + 1) = 0 Then
        peso_itinerario(i) = peso_itinerario(i) + 10
    End If
End If

' /*** 5ª.- No empezar por sexta ***/
If itinerario(i, 0) - cantus_firmus(0) = 9 Then
    peso_itinerario(i) = peso_itinerario(i) + 10
End If

' /*** 5b.- No acabar por sexta ***/
If itinerario(i, 5) - cantus_firmus(5) = 9 Then
    peso_itinerario(i) = peso_itinerario(i) + 10
End If

' /*** 6ª.- No quinta disminuida ***/
If itinerario(i, b) - cantus_firmus(b) = 6 Then
    peso_itinerario(i) = peso_itinerario(i) + 10
End If

' /*** 7ª.- No saltos mayores que el de octava ***/
If Abs(itinerario(i, b + 1) - itinerario(i, b)) > 12 Then
    peso_itinerario(i) = peso_itinerario(i) + 10
End If

'
' /*** 8ª.- No saltos de 7ª ***/
If Abs(itinerario(i, b + 1) - itinerario(i, b)) = 11 Then
    peso_itinerario(i) = peso_itinerario(i) + 10
End If

If Abs(itinerario(i, b + 1) - itinerario(i, b)) = 10 Then
    peso_itinerario(i) = peso_itinerario(i) + 1
End If

Next b
Next i

' /*****
' /***** Reordenamos itinerarios *****/
' /*****

```

```

m = 0

'Print "camino total="; camino_total

For i = 0 To camino_total

If peso_itinerario(i) = 0 Then

    For b = 0 To 5

        itinerario(m, b) = itinerario(i, b)

    Next b

    peso_itinerario(m) = peso_itinerario(i)

    m = m + 1
End If

Next i

'Print "caminos="; m

caminos = m

' /*****
' /*** Rellenamos los tiempos débiles ***
' /*****
Dim itinerario2(6) As Integer
    For i = 0 To caminos - 1
        For b = 0 To 5
            itinerario2(b) = itinerario(i, b)
        Next b
        For b = 0 To 5
            itinerario(i, b * 2) = itinerario2(b)
            itinerario(i, (b * 2) + 1) = 0
        Next b
    Next i

' /*****
' /* REGLAS PARA TIEMPOS DÉBILES */
' /*****
h = 0
p = 1
For i = 0 To caminos - 1
    For b = 0 To 12 Step 2

        ' /*****
        ' /* 1.-Nota de paso
        ' /*****

        ' /* Tercera menor

If (itinerario(i, b + 2) - itinerario(i, b) = 3) Then
    itinerario(i, b + 1) = itinerario(i, b) + 1
    For n = 0 To 18

```

```

        If itinerario(i, b + 1) = notas_posib(n) Then p = 0
    Next n
    itinerario(i, b + 1) = itinerario(i, b + 1) + p
    p = 1
End If

    If (itinerario(i, b) - itinerario(i, b + 2) = 3) Then
        itinerario(i, b + 1) = itinerario(i, b) - 1
        For n = 0 To 19
            If itinerario(i, b + 1) = notas_posib(n) Then p = 0
        Next n
        itinerario(i, b + 1) = itinerario(i, b + 1) - p
        p = 1

    End If

    /* Tercera Mayor

    If (itinerario(i, b + 2) - itinerario(i, b) = 4) Then
        itinerario(i, b + 1) = itinerario(i, b) + 2
    End If

    If (itinerario(i, b) - itinerario(i, b + 2) = 4) Then
        itinerario(i, b + 1) = itinerario(i, b) - 2
    End If

    /******
    /* Floreo superior
    /******
    p = 1
    If (itinerario(i, b) = itinerario(i, b + 2)) Then
        itinerario(i, b + 1) = itinerario(i, b) + 1
        For n = 0 To 19
            If itinerario(i, b + 1) = notas_posib(n) Then p = 0
        Next n
        itinerario(i, b + 1) = itinerario(i, b + 1) + p
        p = 1
    End If

    Next b
    Next i

    /*** Seleccionamos itinerarios con al menos 8 notas puestas
    m = 0
    For i = 0 To caminos - 1
        For b = 0 To 11
            If itinerario(i, b) = 0 Then peso_itinerario(i) =
peso_itinerario(i) + 0.25
        Next b
        If peso_itinerario(i) < 1 Then /*** LÍMITE ***/

        For b = 0 To 11
            itinerario(m, b) = itinerario(i, b)
            'Print itinerario(m, b); " ";
        Next b
        'Print
        peso_itinerario(m) = peso_itinerario(i)
    
```

```

        m = m + 1
    End If

    Next i

    caminos = m

    '/***** Eliminaamos faltas de los caminos resultantes *****/
    For i = 0 To caminos - 1

        For b = 0 To 11

            ' /*** 1ª.- Eliminamos quintas paralelas ***/

            If itinerario(i, b) - cantus_firmus(Int(b / 2)) = 7 Then '/*
caminos con 5ªs */

                If itinerario(i, b + 1) - cantus_firmus(Int(b / 2) + 1) = 7 Then
                    peso_itinerario(i) = peso_itinerario(i) + 10

                End If
            End If

            ' /*** 2ª.- Eliminamos octavas paralelas ***/

            If itinerario(i, b) - cantus_firmus(Int(b / 2)) = 12 Then '/*
caminos con octavas */

                If itinerario(i, b + 1) - cantus_firmus(Int(b / 2) + 1) = 12
Then
                    peso_itinerario(i) = peso_itinerario(i) + 10

                End If
            End If

            ' /*** 3ª.- Eliminamos la duplicación de la sensible

                If itinerario(i, b) = 59 Or itinerario(i, b) = 71 Then
                    If cantus_firmus(Int(b / 2)) = 59 Or cantus_firmus(Int(b / 2))
= 71 Then
                        peso_itinerario(i) = peso_itinerario(i) + 10
                    End If
                End If

            Next b
        Next i

        ' /*** Reordenamos
        m = 0
        For i = 0 To caminos - 1

            If peso_itinerario(i) < 1 Then

```

```

        For b = 0 To 11
            itinerario(m, b) = itinerario(i, b)
        Next b
        peso_itinerario(m) = peso_itinerario(i)
        m = m + 1
    End If

Next i
caminos = m

' /*****
' /***** Completamos los caminos *****/
' /*****
m = 0
p = 1
For i = 0 To caminos - 1
    For b = 0 To 10 Step 2
        If itinerario(i, b + 1) = 0 Then

            If itinerario(i, b) = cantus_firmus(Int(b / 2)) Then '/* UNÍSONO
                itinerario(i, b + 1) = cantus_firmus(Int(b / 2)) + 3
                For n = 0 To 18
                    If itinerario(i, b + 1) = notas_posib(n) Then p = 0
                Next n
                itinerario(i, b + 1) = itinerario(i, b + 1) + p
                p = 1
            End If

            If itinerario(i, b) = cantus_firmus(Int(b / 2)) - 3 Then '/* 3ªm
des
                itinerario(i, b + 1) = cantus_firmus(Int(b / 2)) + 7
            End If

            If itinerario(i, b) = cantus_firmus(Int(b / 2)) - 4 Then '/*
3ªM desc
                itinerario(i, b + 1) = cantus_firmus(Int(b / 2)) + 7
            End If

            If itinerario(i, b) = cantus_firmus(Int(b / 2)) + 8 Then '/*
6ªm asc
                itinerario(i, b + 1) = cantus_firmus(Int(b / 2)) + 3
            End If

            If itinerario(i, b) = cantus_firmus(Int(b / 2)) + 9 Then '/*
6ªM asc
                itinerario(i, b + 1) = cantus_firmus(Int(b / 2)) + 4
            End If

            If itinerario(i, b) = cantus_firmus(Int(b / 2)) + 3 Then '/*
3ªm asc
                itinerario(i, b + 1) = cantus_firmus(Int(b / 2)) + 8
            End If

            If itinerario(i, b) = cantus_firmus(Int(b / 2)) + 4 Then '/* 3ªM
asc
                itinerario(i, b + 1) = cantus_firmus(Int(b / 2)) + 7
            End If

            If itinerario(i, b) = cantus_firmus(Int(b / 2)) + 7 Then '/* 5ªJ
asc
                itinerario(i, b + 1) = cantus_firmus(Int(b / 2)) + 3
                For n = 0 To 18

```

```

        If itinerario(i, b + 1) = notas_posib(n) Then p = 0
    Next n
        itinerario(i, b + 1) = itinerario(i, b + 1) + p
        p = 1
    End If

    If itinerario(i, b) = cantus_firmus(Int(b / 2)) + 12 Then '/*
8ªJ asc
        itinerario(i, b + 1) = cantus_firmus(Int(b / 2)) + 7
    End If

    If itinerario(i, b) = cantus_firmus(Int(b / 2)) + 15 Then '/*
10ªm asc
        itinerario(i, b + 1) = cantus_firmus(Int(b / 2)) + 12
    End If
    If itinerario(i, b) = cantus_firmus(Int(b / 2)) + 16 Then '/* 10ªM
asc
        itinerario(i, b + 1) = cantus_firmus(Int(b / 2)) + 12
    End If

End If

Next b
Next i

' /*****
' /***** Reglas finales de selección *****/
' /*****

For i = 0 To caminos - 1          '/* Ponemos los pesos a 0
    peso_itinerario(i) = 0
Next i

For i = 0 To caminos - 1
    For b = 0 To 11

        ' /*** 10ª.- No saltos de 7ª ***/
        If Abs(itinerario(i, b + 1) - itinerario(i, b)) = 11 Then
            peso_itinerario(i) = peso_itinerario(i) + 1
        End If

        If Abs(itinerario(i, b + 1) - itinerario(i, b)) = 10 Then
            peso_itinerario(i) = peso_itinerario(i) + 1
        End If

        ' /*** 10ª.- No saltos de 9ª ***/
        If Abs(itinerario(i, b + 1) - itinerario(i, b)) = 13 Then
            peso_itinerario(i) = peso_itinerario(i) + 1
        End If

        If Abs(itinerario(i, b + 1) - itinerario(i, b)) = 14 Then
            peso_itinerario(i) = peso_itinerario(i) + 1
        End If

        ' /*** 10ª.- No empezar por 5ª ***/
        If Abs(itinerario(i, 0) - cantus_firmus(0)) = 7 Then
            peso_itinerario(i) = peso_itinerario(i) + 1
        End If

        ' /*** 10ª.- No acabar por 5ª ***/
        If Abs(itinerario(i, 10) - cantus_firmus(0)) = 7 Then

```

```

    peso_itinerario(i) = peso_itinerario(i) + 1
End If

'/** 2ba.- Eliminamos octavas paralelas ***/

    If itinerario(i, b + 1) - cantus_firmus(Int(b / 2)) = 12 Then
'/* caminos con octavas */

        If itinerario(i, b + 3) - cantus_firmus(Int(b / 2) + 1) = 12
Then
            peso_itinerario(i) = peso_itinerario(i) + 1

        End If
    End If

'/** 2a.- Eliminamos octavas paralelas ***/

    If itinerario(i, b) - cantus_firmus(Int(b / 2)) = 12 Then '/*
caminos con octavas */

        If itinerario(i, b + 1) - cantus_firmus(Int(b / 2) + 1) = 12
Then
            peso_itinerario(i) = peso_itinerario(i) + 10

        End If
    End If

'/** 2a.- Eliminamos quintas paralelas ***/

    If itinerario(i, b + 1) - cantus_firmus(Int(b / 2)) = 7 Then
'/* caminos con octavas */

        If itinerario(i, b + 3) - cantus_firmus(Int(b / 2) + 1) = 7
Then
            peso_itinerario(i) = peso_itinerario(i) + 1

        End If
    End If

'/** 2a.- Eliminamos quintas paralelas ***/

    If itinerario(i, b) - cantus_firmus(Int(b / 2)) = 7 Then '/*
caminos con octavas */

        If itinerario(i, b + 1) - cantus_firmus(Int(b / 2) + 1) = 7 Then
            peso_itinerario(i) = peso_itinerario(i) + 1

        End If
    End If

'/** 3a.- Eliminamos la duplicación de la sensible

    If itinerario(i, b) = 59 Or itinerario(i, b) = 71 Then
        If cantus_firmus(Int(b / 2)) = 59 Or cantus_firmus(Int(b / 2))
= 71 Then
            peso_itinerario(i) = peso_itinerario(i) + 1
        End If
    End If

'/** 4a.- Penúltimo compás

```

```

        If cantus_firmus(4) = 62 Then
            If itinerario(i, 8) <> 69 Then peso_itinerario(i) =
peso_itinerario(i) + Cadencia.Value / 100
            If itinerario(i, 9) <> 71 Then peso_itinerario(i) =
peso_itinerario(i) + Cadencia.Value / 100
            If itinerario(i, 10) <> 72 Then peso_itinerario(i) =
peso_itinerario(i) + Cadencia.Value / 100
            End If

'/** 5ª.- Nota repetida

        If itinerario(i, b) = itinerario(i, b + 1) Then
            peso_itinerario(i) = peso_itinerario(i) + NotaRep.Value / 100
            End If

'/** 6ª.- Giros repetidos

        If itinerario(i, b) = itinerario(i, b + 2) Then
            If itinerario(i, b + 1) = itinerario(i, b + 3) Then
                peso_itinerario(i) = peso_itinerario(i) + GirosRep.Value / 100
            End If
        End If

'/** Comprobamos notas posibles
        p = 1
        For n = 5 To 17
            If itinerario(i, b) = notas_posib(n) Then p = 0
        Next n
        If p = 1 Then peso_itinerario(i) = peso_itinerario(i) + 1
        p = 1

    Next b
    Next i

'/** Reordenamos
    m = 0
    For i = 0 To caminos - 1

        If peso_itinerario(i) < Límite.Value / 10 Then

            For b = 0 To 11
                itinerario(m, b) = itinerario(i, b)
            Next b
            peso_itinerario(m) = peso_itinerario(i)
            m = m + 1
        End If

    Next i
    caminos = m - 1

If caminos > 0 Then
    Unload Pentagramas2
    Load Pentagramas2
    Pentagramas2.Show
End If

```

```

End Sub

Private Sub Command2_Click()
SegundaEsp.Hide
End Sub

Private Sub Límite_Change()
Lim.Text = Format((Límite.Value / 10), "0.00")
End Sub

Private Sub TercerasSeg_Scroll()
Text2.Text = Format(TercerasSeg.Value / 100, "0.00")
End Sub

Private Sub TritonoM_Scroll()
Text1.Text = Format(TritonoM.Value / 100, "0.00")
End Sub

Private Sub Unís_Change()
Text3.Text = Format(Unís.Value / 100, "0.00")
End Sub

Private Sub Unís_Scroll()
Text3.Text = Format(Unís.Value / 100, "0.00")
End Sub

Private Sub GirosRep_Change()
Text9.Text = Format(GirosRep.Value / 100, "0.00")
End Sub

Private Sub NotaRep_Change()
Text7.Text = Format(NotaRep.Value / 100, "0.00")
End Sub

```

5.-Pentagramas2(Pentagramas2.Frm)

```

Private Sub Armon_Change()
Line8(0).Visible = False
Line8(1).Visible = False
Line8(2).Visible = False
Line8(3).Visible = False
Line8(4).Visible = False
Line8(5).Visible = False
Line8(6).Visible = False

pk = 0
p = 0
For i = 0 To 9
p = 0

If Module1.itinerario(Armon.Text, i) = 60 Then
    Blanca(i).Visible = True
    Blanca(i).Move 1150 + (i * 500), 900
    Line8(pk).Visible = True

```

```

Line8(pk).X1 = 1080 + (i * 500)
Line8(pk).X2 = 1420 + (i * 500)
Line8(pk).Y1 = 1890
Line8(pk).Y2 = 1890
pk = pk + 1
p = 1
End If
If Module1.itinerario(Armon.Text, i) = 62 Then
    Blanca(i).Visible = True
    Blanca(i).Move 1150 + (i * 500), 840
    p = 1
End If
If Module1.itinerario(Armon.Text, i) = 64 Then
    Blanca(i).Visible = True
    Blanca(i).Move 1150 + (i * 500), 735
    p = 1
End If
If Module1.itinerario(Armon.Text, i) = 65 Then
    Blanca(i).Visible = True
    Blanca(i).Move 1150 + (i * 500), 630
    p = 1
End If
If Module1.itinerario(Armon.Text, i) = 67 Then
    Blanca(i).Visible = True
    Blanca(i).Move 1150 + (i * 500), 535
    p = 1
End If
If Module1.itinerario(Armon.Text, i) = 69 Then
    Blanca(i).Visible = True
    Blanca(i).Move 1150 + (i * 500), 460
    p = 1
End If
If Module1.itinerario(Armon.Text, i) = 71 Then
    Blanca(i).Visible = True
    Blanca(i).Move 1150 + (i * 500), 410
    p = 1
End If
    If Module1.itinerario(Armon.Text, i) = 72 Then
        Blanca(i).Visible = True
        Blanca(i).Move 1150 + (i * 500), 315
        p = 1
    End If
If Module1.itinerario(Armon.Text, i) = 74 Then
    Blanca(i).Visible = True
    Blanca(i).Move 1150 + (i * 500), 210
    p = 1
End If
If Module1.itinerario(Armon.Text, i) = 76 Then
    Blanca(i).Visible = True
    Blanca(i).Move 1150 + (i * 500), 115
    p = 1
End If
If Module1.itinerario(Armon.Text, i) = 77 Then
    Blanca(i).Visible = True
    Blanca(i).Move 1150 + (i * 500), 10
    p = 1
End If
If Module1.itinerario(Armon.Text, i) = 79 Then
    Blanca(i).Visible = True
    Blanca(i).Move 1150 + (i * 500), -80
    p = 1
End If
If Module1.itinerario(Armon.Text, i) = 81 Then

```

```

        Blanca(i).Visible = True
        Blanca(i).Move 1150 + (i * 500), -160
        p = 1
        End If
    If p = 0 Then
        Blanca(i).Visible = False
        End If

    Next i

    i = 10
    p = 0
    If Module1.itinerario(Armon.Text, i) = 60 Then
        Redonda(6).Visible = True

        Redonda(6).Move 1150 + (i * 500), 900
        Line8(pk).Visible = True
        Line8(pk).X1 = 1080 + (i * 500)
        Line8(pk).X2 = 1420 + (i * 500)
        Line8(pk).Y1 = 1890
        Line8(pk).Y2 = 1890

        p = 1
        End If

    If Module1.itinerario(Armon.Text, i) = 64 Then
        Redonda(6).Visible = True
        Redonda(6).Move 1150 + (i * 500), 735
        p = 1
        End If
    If Module1.itinerario(Armon.Text, i) = 67 Then
        Redonda(6).Visible = True
        Redonda(6).Move 1150 + (i * 500), 535
        p = 1
        End If

        If Module1.itinerario(Armon.Text, i) = 72 Then
            Redonda(6).Visible = True
            Redonda(6).Move 1150 + (i * 500), 315
            p = 1
            End If

    If Module1.itinerario(Armon.Text, i) = 76 Then
        Redonda(6).Visible = True
        Redonda(6).Move 1150 + (i * 500), 115
        p = 1
        End If
    If Module1.itinerario(Armon.Text, i) = 79 Then
        Redonda(6).Visible = True
        Redonda(6).Move 1150 + (10 * 500), -80
        p = 1
        End If

    If p = 0 Then
        Redonda(6).Visible = False
        End If

End Sub

Private Sub Command1_Click()
Pentagramas.Hide
End Sub

```

```

Private Sub Form_Load()
'Tabla de conversión para cantus firmus

For i = 0 To 5
If Module1.cantus_firmus(i) = 52 Then Redonda(i).Move 1150 + (i *
1000), 2415
If Module1.cantus_firmus(i) = 53 Then Redonda(i).Move 1150 + (i *
1000), 2310
If Module1.cantus_firmus(i) = 55 Then Redonda(i).Move 1150 + (i *
1000), 2200
If Module1.cantus_firmus(i) = 57 Then Redonda(i).Move 1150 + (i *
1000), 2125
If Module1.cantus_firmus(i) = 59 Then Redonda(i).Move 1150 + (i *
1000), 2050
If Module1.cantus_firmus(i) = 60 Then Redonda(i).Move 1150 + (i *
1000), 1980
If Module1.cantus_firmus(i) = 62 Then Redonda(i).Move 1150 + (i *
1000), 1890
If Module1.cantus_firmus(i) = 64 Then Redonda(i).Move 1150 + (i *
1000), 1785
If Module1.cantus_firmus(i) = 65 Then Redonda(i).Move 1150 + (i *
1000), 1680
If Module1.cantus_firmus(i) = 67 Then Redonda(i).Move 1150 + (i *
1000), 1600
If Module1.cantus_firmus(i) = 69 Then Redonda(i).Move 1150 + (i *
1000), 1500
Next i

'Tabla de conversión para resultado
pk = 0
For i = 0 To 9
Line8(0).Visible = False
Line8(1).Visible = False
Line8(2).Visible = False
Line8(3).Visible = False
Line8(4).Visible = False
Line8(5).Visible = False
Line8(6).Visible = False

If Module1.itinerario(Armon.Text, i) = 60 Then
    Blanca(i).Visible = True
    Blanca(i).Move 1150 + (i * 500), 900
    Line8(pk).Visible = True
    Line8(pk).X1 = 1080 + (i * 500)
    Line8(pk).X2 = 1420 + (i * 500)
    Line8(pk).Y1 = 1890
    Line8(pk).Y2 = 1890
    pk = pk + 1
    p = 1
End If
If Module1.itinerario(0, i) = 62 Then Blanca(i).Move 1150 + (i * 500),
840
If Module1.itinerario(0, i) = 64 Then Blanca(i).Move 1150 + (i * 500),
735
If Module1.itinerario(0, i) = 65 Then Blanca(i).Move 1150 + (i * 500),
630
If Module1.itinerario(0, i) = 67 Then Blanca(i).Move 1150 + (i * 500),
535
If Module1.itinerario(0, i) = 69 Then Blanca(i).Move 1150 + (i * 500),
460

```

```

If Module1.itinerario(0, i) = 71 Then Blanca(i).Move 1150 + (i * 500),
410
If Module1.itinerario(0, i) = 72 Then Blanca(i).Move 1150 + (i * 500),
315
If Module1.itinerario(0, i) = 74 Then Blanca(i).Move 1150 + (i * 500),
210
If Module1.itinerario(0, i) = 76 Then Blanca(i).Move 1150 + (i * 500),
115
If Module1.itinerario(0, i) = 77 Then Blanca(i).Move 1150 + (i * 500),
10
If Module1.itinerario(0, i) = 79 Then Blanca(i).Move 1150 + (i * 500),
-80
If Module1.itinerario(0, i) = 0 Then Blanca(i).Visible = False

Next i
i = 10
p = 0
If Module1.itinerario(Armon.Text, i) = 60 Then
    Redonda(6).Visible = True

    Redonda(6).Move 1150 + (i * 500), 900
    Line8(pk).Visible = True
    Line8(pk).X1 = 1080 + (i * 500)
    Line8(pk).X2 = 1420 + (i * 500)
    Line8(pk).Y1 = 1890
    Line8(pk).Y2 = 1890

    p = 1
End If

If Module1.itinerario(Armon.Text, i) = 64 Then
    Redonda(6).Visible = True
    Redonda(6).Move 1150 + (i * 500), 735
    p = 1
End If

If Module1.itinerario(Armon.Text, i) = 67 Then
    Redonda(6).Visible = True
    Redonda(6).Move 1150 + (i * 500), 535
    p = 1
End If

If Module1.itinerario(Armon.Text, i) = 72 Then
    Redonda(6).Visible = True
    Redonda(6).Move 1150 + (i * 500), 315
    p = 1
End If

If Module1.itinerario(Armon.Text, i) = 76 Then
    Redonda(6).Visible = True
    Redonda(6).Move 1150 + (i * 500), 115
    p = 1
End If

If Module1.itinerario(Armon.Text, i) = 79 Then
    Redonda(6).Visible = True
    Redonda(6).Move 1150 + (10 * 500), -80
    p = 1
End If

If p = 0 Then
    Redonda(6).Visible = False
End If

```

End Sub

```
Private Sub HScroll1_Change()  
Armon.Text = HScroll1.Value  
If HScroll1.Value > caminos - 1 Then Armon.Text = caminos - 1  
HScroll1.Max = caminos - 1  
Picture1.Cls  
For i = 0 To 10  
Picture1.Print itinerario(HScroll1.Value, i); " ";  
Next i  
End Sub
```

```
Private Sub HScroll1_Scroll()  
Armon.Text = HScroll1.Value  
If HScroll1.Value > caminos - 1 Then Armon.Text = caminos - 1  
HScroll1.Max = caminos - 1  
  
End Sub
```

```
Private Sub Play_Click()  
Open "datos.txt" For Output As #1  
Print #1, "MFile 0 1 24"  
Print #1, "MTrk"  
Print #1, "0 Tempo 300000"  
'Nota  
Print #1, "0 On ch=1 n="; Module1.itinerario(Armon.Text, 0); " v=64"  
Print #1, "0 On ch=2 n="; Module1.cantus_firmus(0); " v=64"  
Print #1, "48 On ch=1 n="; Module1.itinerario(Armon.Text, 0); " v=0"  
Print #1, "48 On ch=1 n="; Module1.itinerario(Armon.Text, 1); "  
v=64"  
Print #1, "96 On ch=1 n="; Module1.itinerario(Armon.Text, 1); " v=0"  
Print #1, "96 On ch=2 n="; Module1.cantus_firmus(0); " v=0"  
'Nota  
Print #1, "96 On ch=1 n="; Module1.itinerario(Armon.Text, 2); "  
v=64"  
Print #1, "96 On ch=2 n="; Module1.cantus_firmus(1); " v=64"  
Print #1, "144 On ch=1 n="; Module1.itinerario(Armon.Text, 2); "  
v=0"  
Print #1, "144 On ch=1 n="; Module1.itinerario(Armon.Text, 3); "  
v=64"  
Print #1, "192 On ch=1 n="; Module1.itinerario(Armon.Text, 3); " v=0"  
Print #1, "192 On ch=2 n="; Module1.cantus_firmus(1); " v=0"  
'Nota  
Print #1, "192 On ch=1 n="; Module1.itinerario(Armon.Text, 4); " v=64"  
Print #1, "192 On ch=2 n="; Module1.cantus_firmus(2); " v=64"  
Print #1, "240 On ch=1 n="; Module1.itinerario(Armon.Text, 4); "  
v=0"  
Print #1, "240 On ch=1 n="; Module1.itinerario(Armon.Text, 5); "  
v=64"  
Print #1, "288 On ch=1 n="; Module1.itinerario(Armon.Text, 5); " v=0"  
Print #1, "288 On ch=2 n="; Module1.cantus_firmus(2); " v=0"  
'Nota  
Print #1, "288 On ch=1 n="; Module1.itinerario(Armon.Text, 6); " v=64"  
Print #1, "288 On ch=2 n="; Module1.cantus_firmus(3); " v=64"  
Print #1, "336 On ch=1 n="; Module1.itinerario(Armon.Text, 6); "  
v=0"  
Print #1, "336 On ch=1 n="; Module1.itinerario(Armon.Text, 7); "  
v=64"  
Print #1, "384 On ch=1 n="; Module1.itinerario(Armon.Text, 7); "  
v=0"  
Print #1, "384 On ch=2 n="; Module1.cantus_firmus(3); " v=0"  
'Nota  
Print #1, "384 On ch=1 n="; Module1.itinerario(Armon.Text, 8); " v=64"
```

```

Print #1, "384 On ch=2 n="; Module1.cantus_firmus(4); " v=64"
Print #1, "432 On ch=1 n="; Module1.itinerario(Armon.Text, 8); " v=0"
Print #1, "432 On ch=1 n="; Module1.itinerario(Armon.Text, 9); "
v=64"
Print #1, "480 On ch=1 n="; Module1.itinerario(Armon.Text, 9); " v=0"
Print #1, "480 On ch=2 n="; Module1.cantus_firmus(4); " v=0"
'Nota
Print #1, "480 On ch=1 n="; Module1.itinerario(Armon.Text, 10); "
v=64"
Print #1, "480 On ch=2 n="; Module1.cantus_firmus(5); " v=64"
Print #1, "576 On ch=1 n="; Module1.itinerario(Armon.Text, 1); " v=0"
Print #1, "576 On ch=2 n="; Module1.cantus_firmus(5); " v=0"
'Nota
Print #1, "576 Meta TrkEnd"
Print #1, "TrkEnd"
Close #1

```

```

'Ejecutamos T2mf

```

```

Dim hWndAct As Long, vr, hr

hWndAct = GetActiveWindow()
vr = Shell("t2mf -r datos.txt v2.mid")
Do While GetActiveWindow() = hWndAct
    vr = DoEvents()
Loop
hWndAct = GetForegroundWindow()
Do While IsWindow(hWndAct)
    vr = DoEvents()
Loop

hr = Shell("mplayer2 v2.mid")

```

```

End Sub

```

```

Private Sub Text1_Change()
End Sub

```

7.- Bibliografía.

- [Zamacois 94] Joaquín Zamacois, “Tratado de armonía”, Ed. Labor, S.A.
- [Motte 76] Diether de la Motte, “Armonía”, Ed. Labor, S.A.
- [Salzer 52] Felix Salzer, Audición estructural, Ed Labor, S.A.
- [Motte 81] Diether de la Motte, “Contrapunto”, Ed. Labor, S.A.
- [Forner 79] Johannes Forner, Jürgen Wilbrandt, “Contrapunto creativo”, Ed. Labor, S.A.
- [Schoenberg 63] Arnold Schoenberg, “Ejercicios preliminares de contrapunto”, Ed. Labor, S.A.
- [Schoenberg 22] Arnold Schoenberg, “Armonía”, Ed. Real Musical, Madrid.
- [Hoppin 78] Richard H. Hoppin, “La música medieval”, Ed. Akal Música.
- [Ebcioglu 92] Kemal Ebcioglu, Mira Balaban, Otto Laske, “Understanding Music with AI: Perspectives on Music Cognition”.The AAAI Press.
- [Smith 93] Matt Smith, Alan Smail, Geraint A. Wiggins. “Music Education: An Artificial Intelligence Approach” Ed. Springer-Verlag.
- [Minski 86] M. Minski. “The society of Mind”. New York.
- [Lincoln 70] H.B. Lincoln. “The Computer and Music”, Cornell University Press, 1970

- [Smoliar 80] S.W. Smoliar. “”A Computer Aid for Schenkerian Analysis” Computer Music Journal, 4, Summer 1980.
- [Mira 95] J. Mira, A.E. Delgado, J.G. Boticario, F.J. Díez “Aspectos básicos de la inteligencia artificial”. Ed Sanz y Torres.
- [Atkinson 93] Lee Atkinson, Mark Atkinson. “Programación en Borland C++”.Ed. Anaya multimedia.
- [Ceballos 91] F.J. Ceballos. “Enciclopedia del lenguaje C”. Ed. Ra-Ma.
- [Ceballos 96] F.J. Ceballos. “Enciclopedia de Visual Basic”. Ed. Ra-Ma
- [Balfagón 94] Albert Balfagón, Pere Ripoll. “Visual Basic. Programación avanzada en Windows”. Ed Inforbook’s.